Seeing The Light: Artificial Evolution, Real Vision

Inman Harvey and Phil Husbands and Dave Cliff
School of Cognitive and Computing Sciences
University of Sussex, Brighton BN1 9QH, UK
inmanh or philh or davec@cogs.susx.ac.uk

Abstract

This paper describes results from a specialised piece of visuo-robotic equipment which allows the artificial evolution of control systems for visually guided autonomous agents acting in the real world. Preliminary experiments with the equipment are described in which dynamical recurrent networks and visual sampling morphologies are concurrently evolved to allow agents to robustly perform simple visually guided tasks. Some of these control systems are shown to exhibit a surprising degree of adaptiveness when tested against generalised versions of the task for which they were evolved.

1 Introduction

In previous papers (see e.g. [1]) we have discussed our reasons for adopting an evolutionary methodology for the design of control systems for mobile robots using low-bandwidth vision for simple navigational tasks. We also discussed what class of control systems are appropriate for evolutionary development, proposing dynamic recurrent real-time (artificial) neural networks as one strong contender.

The evolutionary process, based on a genetic algorithm [3], involves evaluating, over many generations, whole populations of control systems specified by artificial genotypes. These are interbred using a Darwinian scheme in which the fittest individuals are most likely to produce offspring. Fitness is measured in terms of how good an agent's behaviour is according to some evaluation criterion. The work reported here forms part of a long-term study to explore the viability of such an approach in developing interesting adaptive behaviours in visually guided autonomous robots, and, through analysis, in better understanding general mechanisms underlying the generation of such behaviours.

In this paper we present results from experiments in which visually guided behaviours are artificially evolved in the real world. As far as we know, this is the first time this has been achieved.

2 From Simulation to Reality

The experiments described in earlier papers [1] used simulations of a round two-wheeled mobile robot with touch sensors and just two visual inputs — simulated photoreceptors, with (genetically specified) angles of acceptance, and of eccentricity relative to the frontal direction of the robot. The environment was a simulated circular arena, with black walls and white floor and ceiling; ray-tracing techniques allowed the calculation of visual inputs. Success was reported in evolving control systems (and visual morphologies) which allowed the robot to reach the centre of the arena.

These early experiments were intended to test the plausibility of our approach. However, the simulated visual environment was very simple and it was noted that computational costs would increase dramatically as the visual environment became more complex. Indeed, even ignoring computational costs, the plausible modelling of visual inputs in such circumstances is highly problematic. Hence plans were made to perform the whole evolutionary process with a real robot, moving in the real world, and without recourse to simulated vision.

Artificial evolution in the real world requires equipment which allows the automatic evaluation of very large numbers of robot control systems. With navigation tasks, it is useful to have the position and orientation of a robot continually available to an overseeing program responsible for scoring candidate control systems. Of course, this information should not be available in any way to the individual robot control systems. Automatic re-positioning of the robot at fixed or random positions for the start of each trial is also desirable. Rather than imposing a fixed visual sampling morphology, we believe a more powerful approach is to allow the visual morphology to evolve along with the rest of the control system. This establishes a further desired property of the experimental setup.

One solution might involve the parallel evaluation of populations of control systems using a large number of mobile robots, radio links, recharging stations, and the like. In this paper we describe a much cheaper, shorter term, solution we have developed using a specialised piece of visuo-robotic equipment — the gantry-robot.

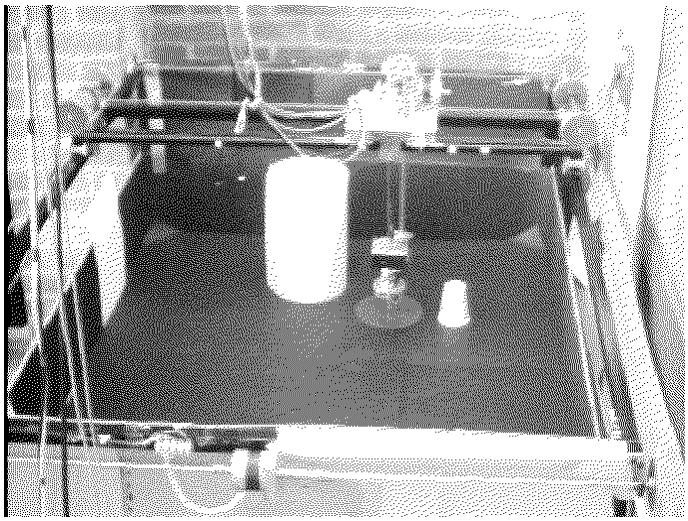

Figure 1: *The Gantry viewed from above. The horizontal girder moves along the side rails, and the robot is suspended from a platform which moves along this girder.*

3 The Gantry-Robot

3.1 Introduction

The gantry-robot can be thought of as occupying a position partway between a physical mobile robot with two wheels and low-bandwidth vision, and the simulation thereof. The robot is physically built, cylindrical, some 150mm in diameter, and moves in a real environment — the term 'robot' is here used to refer to that part which moves around and has the sensors mounted on it. Instead of two wheels, however, the robot is suspended from the gantry-frame with stepper motors that allow translational movement in the X and Y directions, relative to a co-ordinate frame fixed to the gantry (see Figure 1). The maximum X (and Y) speed is about 200mm/s. Such movements, together with appropriate rotation of the sensory apparatus, can be thought of as corresponding to those which would be produced by left and right wheels. The visual sensory apparatus consists of a CCD camera pointing down at a mirror inclined at 45^o to the vertical (see Figure 2). The mirror can be rotated about a vertical axis so that its orientation always corresponds to the direction the 'robot' is facing. The visual inputs undergo some transformations en route to the control system, described in detail below. The hardware is designed so that these transformations are done completely externally to the processing of the control system. If all the transformations made on the sensory inputs and the motor outputs accurately reflected the physics of a real mobile robot, then, in principle, a control system successfully evolved on the gantry could be transplanted to a mobile robot with two genuine wheels, and with photoreceptors instead of the vision system described below. Such a transplantation has not been attempted, and is not a prime concern of our present work with this apparatus. Indeed, there are current limitations, discussed later, which would probably hinder it. Despite this, the experiments discussed here can be considered as having conditions comparable in complexity and difficulty to those met by a free-running mobile robot; our aim is a fairly general investigation of the artificial evolution of sensorimotor control systems. Of course, the optic array available to the robot is now the real thing.

The control system for the robot is a recurrent dynamic neural net, genetically specified, and in practice simulated on a fast personal computer, the 'Brain PC'. During each robot trial this PC is dedicated solely to running the neural net simulation. It receives any changes in visual input by interrupts from a second dedicated 'Vision PC'. A third (single-board) computer, the SBC, sends interrupts to the Brain PC signalling tactile inputs resulting from the robot bumping into walls or physical obstacles. The only outputs of the control system are motor signals specified by values on particular nodes of the neural network; these values are sent, via interrupts, to the SBC, which generates the appropriate stepper motor movements on the gantry.

Thus all interactions between the three computers used (Brain PC, Vision PC and SBC) are mediated by interrupts (see Figure 3); and the overall system is deliberately designed so that these interrupts, although inherently asynchronous and unpredictable, are nevertheless sufficiently infrequent for them not to clash with the intrinsic timescales of the neural network, vision and stepper motor processing.

This setup, with off-board computing and avoidance of tangled umbilicals, means that the apparatus can be run continuously for long periods of time – making artificial evolution feasible. A top-level program automatically

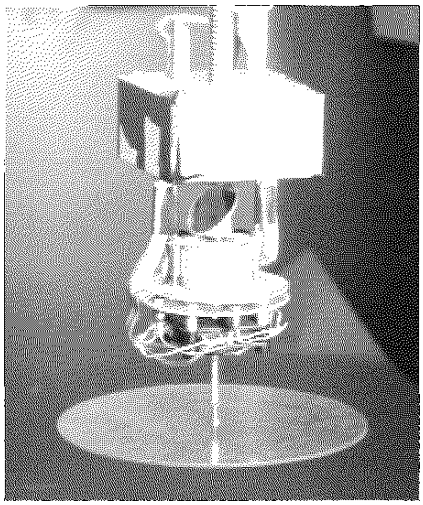

Figure 2: *The gantry-robot. The camera inside the top box points down at the inclined mirror, which can be turned by the stepper-motor beneath. The lower plastic disk is suspended from a joystick, to detect collisions with obstacles.*

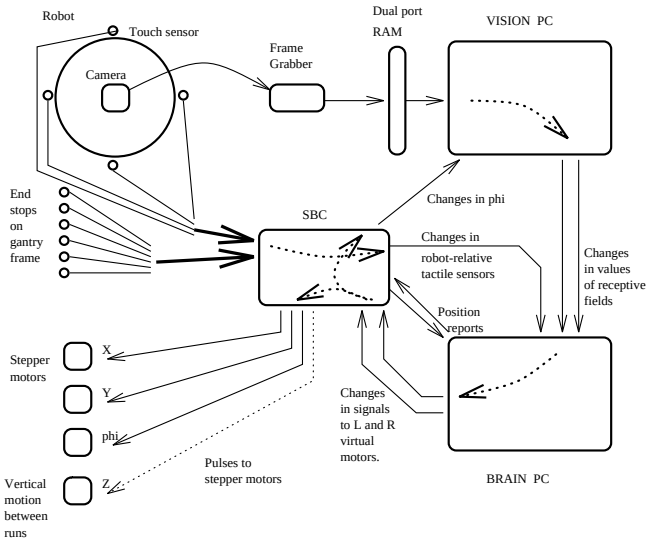

Figure 3: *The different rôles of the Vision computer, the Brain computer and the SBC.*

evaluates, in turn, each member of a population of control systems. A new population is produced by selective interbreeding and the cycle repeats.

3.2 The Vision System

Continuous visual data is derived from the output of a small monochrome CCD camera. With a wide-angle (about 40^o) fixed-focus lens about 6mm in diameter, this is housed in a box facing vertically downwards onto the angled mirror of the robot. The CCD produces composite video output of some 1 volt peak to peak, with a video bandwidth of 4MHz. A purpose-built Frame-Grabber transfers a 64×64 image at 50Hz into a high-speed 2K CMOS dual-port RAM, completely independently and asynchronously relative to any processing of the image by the Vision PC.

We advocate an incremental evolutionary approach, progressing from the simple to the complex. In keeping with this philosophy, current experiments use very low bandwidth vision. This implies sub-sampling the image produced by the camera. Rather than imposing a fixed way of sampling the image, we allow this to evolve along with the neural networks. This is achieved by genetically specifying the size and position of visual receptive fields. These are circular patches within the visual field of the camera (see Figure 4). Up to 256 such receptive fields can be specified with, to 8-bit accuracy: the diameter of the field; and the polar coordinates of the centre of the field relative to the centre of the camera's field of view. The angle of acceptance of the CCD camera (via the mirror) is about 60^o; the maximum angle of acceptance of a receptive field is about 16^o, and its maximum angle of eccentricity off the cameras visual axis is about 22^o. To calculate the signal from such a field, the average is taken of 25 pixels in the camera image scattered across the appropriate area. In this way a value (4 bits) can be calculated for each receptive field at least as fast as the camera image is updated. The only visual inputs available to the genetically designed robot control system are such scalar values.

The Vision PC is dedicated solely to processing the camera output to calculate the visual signals from the receptive fields. At the beginning of a set of trials for a particular robot, the genetic specification for the visual morphology (positions and sizes of receptive fields) is passed to this PC. During each trial, whenever the orientation of the robot changes (the full circle is discretized into 96 orientations) a single byte is sent to the Vision PC from the SBC specifying the new orientation. Whenever the visual input to any of the receptive fields changes in value (scaled in the range 0 to 15) then the details of such a change are sent as single-byte interrupts to the Brain PC.

3.3 The Brain PC

This is a 66MHz 486 PC which has two separate groups of tasks to do at different times. Firstly, the Genetic Algorithm (GA) code is run on this machine. Reproduction, crossover and mutation are performed here in between generations, and at the start of a set of trials for each robot architecture the specification of the visual morphology is transmitted to the Vision PC. As with most GAs, however, the amount of time spent running the genetic machinery is trivial compared with the time spent running the evaluations, and this latter constitutes the second group of tasks.

During an individual evaluation, the Brain PC is ded-

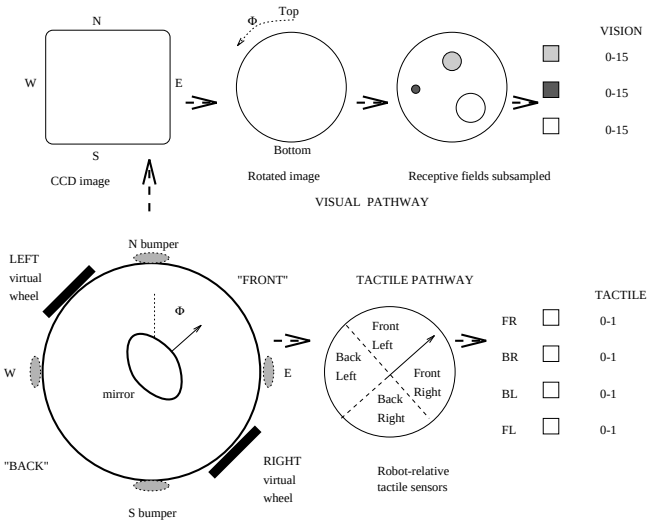

Figure 4: *A schematic of the sensory pathways, visual and tactile, from the gantry-robot hardware at bottom left through to the nodes of the neural network on the right.*

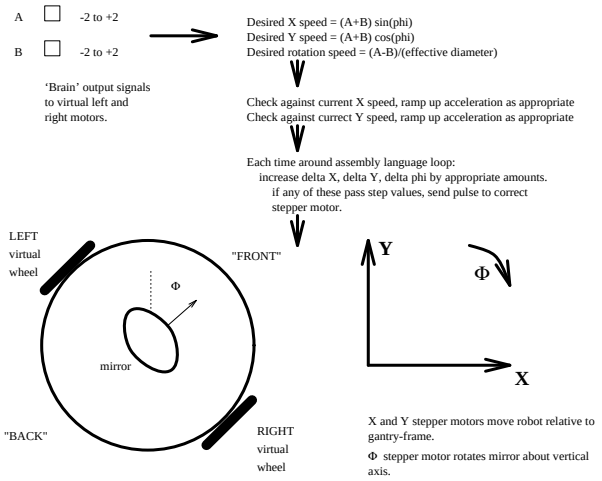

Figure 5: *The conversion from the motor outputs signalled by the neural network to the stepper motor movements which move the gantry-robot.*

icated to running a genetically specified neural network for a fixed period. At intervals during an evaluation, a signal is sent from the Brain PC to the SBC requesting the current position and orientation of the robot. These are used in keeping score according to the current fitness function. At the end of a run, a byte sent from the Brain PC to the SBC requests the return of the robot to the 'origin' of the gantry. The Brain PC receives signals, to be fed into the neural net, representing sensory inputs from the Vision PC and the SBC. The only signals that the Brain PC sends out indicate changes in values of the left and right virtual motors of the robot. These values, which are restricted to integers from -2 to 2, are passed on as single-byte interrupts to the SBC (see Figure 5).

3.4 The SBC

The SBC is a minimal 16 bit 68000 system with 256K of RAM and 128K of ROM, running at 10MHz. It has memory mapped ports that connect it to the Vision PC, the Brain PC and the various switches and motors attached to the gantry. The SBC does all the transformations between hardware-relative and robot-relative signals, plus some housekeeping.

Occasional interrupts from the Brain PC will notify new values of the desired speeds of the virtual left and right wheels. These are translated into desired speeds in the gantry X and Y directions, and desired angular velocity of the mirror. The SBC also keeps track of the current position and orientation of the robot. Instantaneous changes in desired speed cannot be translated directly into instantaneous changes in stepper motor pulse frequency, due to the momentum of the masses these motors must move. Hence speeds are ramped up relatively slowly towards the desired speed — from zero to full speed in about 2 seconds — decelerations are ramped down rather more swiftly. As the mirror is so light, such ramping was not deemed necessary for rotation — with unexpected side-effects described below.

Signals from end-stops for maximum movement along the gantry-frame, and signals from the touch-sensors on the robot, are also processed by the SBC. A plastic disc in the horizontal plane is suspended on a joystick vertically below the robot(see Figure 2). This detects contacts, on each of 4 sides of the robot (in gantry-relative coordinates). The SBC converts these into robot-relative directions.

4 The robot dynamics

Some issues relating to the physical dynamics have already been mentioned; the ramping up and down of stepper motor movements broadly (and perhaps inaccurately) relates to the momentum of a freely mobile wheeled robot.

With the present setup, on collision with a wall, all further movement into the wall is prevented, as is any translational movement along the wall. Hence, of any desired robot motion, only that component perpendicularly away from the wall is allowed, until contact with the wall is lost. Angular velocity that attempts to turn the robot further in towards the wall is ineffective.

One puzzling phenomenon often observed, particularly in initial randomly generated populations, was that of a robot turning on the spot in a noisy fashion. On reflection, this turned out to be an artefact of the way translational momentum had been implemented in the SBC code, but without angular momentum. This clearly showed how the virtual physics as currently implemented does not accurately reflect the real physics of a free mobile robot.

5 Visual Limitations

The visual inputs are currently subject to various limitations which are worth noting. Firstly, the lower part of the robot body is supported from the upper half with two thin vertical bolts, which come into the field of view when the mirror is facing towards them. These appear as dark bars 2 to 3 pixels wide on the CCD image, and affect the values of any receptive fields sampling from this area. In principle this could directly provide visual information for two fixed directions for the robot to 'face'. In addition, these bars tend to occlude any distant target used in navigation trials. For our early crude experiments this may not be too significant, but it certainly will matter when finer resolution is needed, and these bars produce greater effects than background noise levels. In future work we intend to fit a new head on the gantry which overcomes this problem.

Secondly, the fact that the mirror turns in discrete jumps, of 3.75^o at the moment, means that either the angles of acceptance of the receptive fields, or alternatively the horizontal angle subtended by any significant visual features, should be somewhat greater than 3.75^o. This could be overcome with a finer resolution motor.

Thirdly, the visual inputs are naturally noisy (see section 6.2). The natural variation in daylight, as day progresses into night, causes particular problems. When the gantry was exposed to such variation, it was discovered that evolved systems that worked well in the daytime did not work well under artificial light alone at night-time, and vice versa. Our individual robot systems were evaluated over a period of perhaps 3 minutes only, and hence it is no surprise that robustness against such longterm variations was not achieved. Since the recognition of this problem the gantry has been largely shielded against daylight variations. We intend soon to deliberately vary lighting conditions *within* each robot trial, to try to achieve robustness against such variations.

6 Preliminary Experiments

The following sections describe some initial simple experiments we have carried out, mainly to ascertain how well our methods cope with the move from simulations to the real world. We have begun by exploring primitive visually guided behaviours in static environments, concentrating on target approaching. However, as we shall see, some of the evolved control systems showed surprising degrees of adaptiveness when tested on more general versions of the task they were evolved for.

6.1 Networks and Genotypes

In all of the experiments reported here we used the same networks and genetic encoding schemes as in our earlier simulation work (for full details see [1]). This was mainly because we have a detailed understanding of their properties and wanted to see how well they transferred to real world tasks. However, they are the simplest, and we believe least powerful, of the classes of networks and genetic encodings we advocate, and we are currently exploring more sophisticated methods. Briefly, the evolutionary algorithms search concurrently for a network architecture and visual morphology capable of generating behaviours resulting in a high score on an evaluation function that implicitly describes a visually guided task. This is achieved by using a genetic algorithm acting on pairs of 'chromosomes' encoding the network and visual morphology of a robot control system. One of the chromosomes is a fixed length bit string encoding the position and size of three visual receptive fields as described above. The other is a variable length character string encoding the architecture of the control network. Each net has a fixed number of input nodes and output nodes, one input for each visual receptive field and one for each of the four tactile sensors described earlier. There are four output nodes, two for each 'virtual motor'. The output signals of these pairs are subtracted to give motor signals in the range [-1,1]. The genotypes encode for a variable number of hidden units and for a variable number of unrestricted excitatory and inhibitory connections between the nodes.

The model neurons use separate channels for excitation and inhibition. Real values in the range [0,1] propagate along excitatory links subject to delays associated with the links. The inhibitory (or veto) channel mechanism works as follows. If the sum of excitatory inputs exceeds a threshold, T_v, the value 1.0 is propagated along any inhibitory output links the unit may have, otherwise a value of 0.0 is propagated. Veto links also have associated delays. Any unit that receives a non zero inhibitory input has its excitatory output reduced to zero (i.e. is vetoed). In the absence of inhibitory input, excitatory outputs are produced by summing all excitatory inputs, adding a quantity of noise, and passing the resulting sum through a simple linear threshold function, $F(x)$, given below. Noise was added to provide further potentially interesting and useful dynamics. The noise was uniformly distributed in the real range [-N,+N].

$$F(x) = \begin{cases} 0, & \text{if } x \leq T_1 \\ \frac{x-T_1}{T_2-T_1}, & \text{if } T_1 < x < T_2 \\ 1, & \text{if } x \geq T_2. \end{cases} \quad (1)$$

The networks' continuous nature was modelled by using very fine time slice techniques. In the experiments described in this paper the following neuron parameter setting were used: N=0.1, T_v=0.75, T_1=0.0 and T_2=2.0. The networks are hardwired in the sense that they do not undergo any architectural changes during their lifetime, they all had unit weights and time delays on their connections.

6.2 Experimental Details

In each of the experiments a population size of 30 was used with a genetic algorithm employing a linear rank-based selection method, ensuring the best individual in a population was twice as likely to breed as the median individual. Each generation took about 1.5 hours to evaluate. The most fit individual was always carried over to the next generation unchanged. A specialised crossover allowing small changes in length between offspring and parents was used [1]. Mutation rates were set at 1.0 bit per vision chromosome and 1.8 bits per network chromosome.

With the walls and floor of the gantry environment predominantly dark, initial tasks were navigating towards white paper targets. In keeping with the incremental evolutionary methodology, deliberately simple visual environments are used initially, as a basis to moving on to more complex ones. Illumination was provided by fluorescent lights in the ceiling above, with the gantry screened from significant daylight variations. However, the dark surfaces did not in practice provide uniform light intensities, neither over space nor over time. Even when the robot was stationary, individual pixel values would fluctuate by up to 2 units, on a scale of 0 to 15. Varying illuminance of different parts of the walls provided potential visual information, other than the targets specifically displayed.

6.2.1 Big Target

In the first experiment, one long gantry wall was covered with white paper, to a width of 150cm and a height of 22cm; the mirror on the robot, which effectively determines the position of the visual inputs, came about 2/3 of the way up on this white wall. The evaluation function $\mathcal{E}_1$, to be maximised, implicitly defines a target locating task, which we hoped would be achieved by visuomotor coordination:

$$\mathcal{E}_1 = \sum_{i=1}^{i=20} Y_i \qquad (2)$$

where Y_i are the perpendicular distances of the robot from the wall opposite that to which the target is attached, sampled at 20 fixed time intervals throughout a robot trial which lasted a total of about 25 seconds. The closer to the target the higher the score. For each robot architecture 4 trials were run, each starting in the same distant corner, but facing in 4 different directions; these directions were approximately in 4 different quadrants, to give a range of starts facing into obstacle walls as well as towards the target. As the final fitness of a robot control architecture was based on the *worst* of the 4 trials (to encourage robustness), and since in this case scores accumulated monotonically through a trial, this allowed later trials among the 4 to be prematurely terminated when they bettered previous trials. In addition, any control systems that had not produced any movement by 1/3 of the way into a trial was aborted and given zero score.

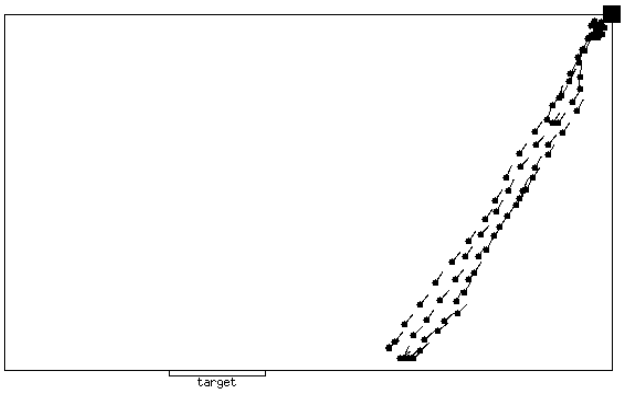

Figure 6: *From those evolved for the first task, this is the behaviour of the one best at the 2nd evaluation function. The dots, and trailing lines, show the front of the robot, and its orientation. Coarsely sampled positions from each of 4 runs are shown, starting in different orientations from the top right corner.*

Two runs starting from a random initial populations made little progress after 15 generations. For reasons described in Section 9, we then tried starting from a converged population made entirely of clones of a single randomly generated individual picked out by us as displaying vaguely interesting behaviour (but by no means able to do anything remotely like locate and approach the target). In two runs using this method very fit individuals appeared in less than 10 generations. From a start close to a corner, they would turn, avoiding contact with the walls by vision alone[1]. The best would rotate on the spot until the target was in their visual field and then move straight towards it, stopping when they reached it.

6.2.2 Small Target

The experiment continued from the stage already reached, but now using a much narrower target (22cm) placed about 2/3 of the way along the same wall the large target had been on, and away from the robot's starting corner (see Figure 6), with evaluation $\mathcal{E}_2$:

$$\mathcal{E}_2 = \sum_{i=1}^{i=20} (-d_i) \qquad (3)$$

where d_i is the distance of the robot from the centre of the target at one of the sampled instances during an evaluation run. Again, the fitness of an individual was set to the worst evaluation score from four runs with starting

[1] They were forced into this by a software error, only discovered later, which meant that all the tactile sensors were turned off. This made this initial task far harder than we had intended.

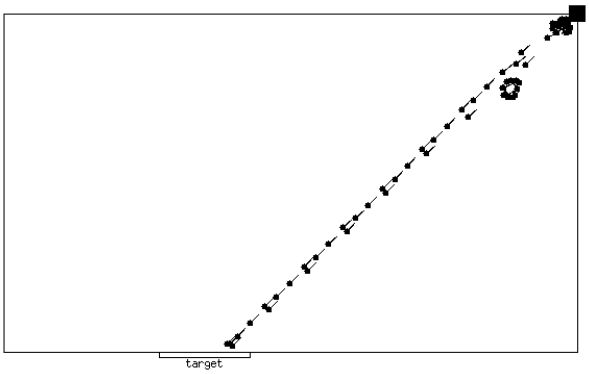

Figure 7: *Behaviour of the best of a later generation evolved under 2nd evaluation function. Format as in previous Figure.*

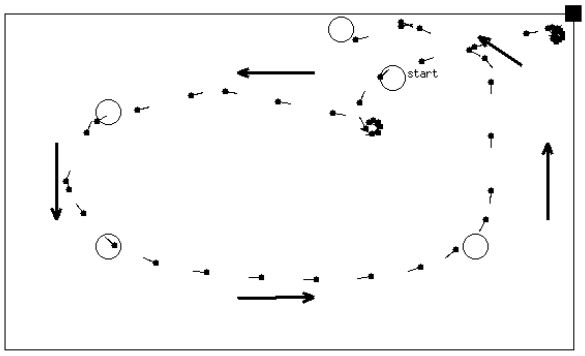

Figure 8: *Tracking behaviour of the control system that generated the behaviour shown in previous Figure. The unfilled circles show the position of the target at a number of points on its path (starting position indicated). The arrows roughly indicate the path of the target.*

conditions as in the first experiment. The initial population used was the 12th generation from a run of the first experiment (i.e. we incrementally evolved on top of the existing behaviours). The behaviour of the best of this initial population is shown in Figure 6. Interestingly, this was not the best at the previous task – that individual did very poorly on the new task.

Within six generations a network architecture and visual morphology had evolved displaying the behaviour shown in Figure 7. This control system was tested from widely varying random starting positions and orientations, with the target in different places, and with smaller and different shaped targets. Its behaviour was general enough to cope with all these conditions for which it had not explicitly been evolved.

For comparison a second evolutionary run using $\mathcal{E}_2$ throughout was undertaken; this time $\mathcal{E}_1$, and the big target, were not used as a stepping stone. The run started from the same initial converged population as was used for the first task. High scoring individuals emerged after 15 generations. When tested on more general versions of the task they performed much worse than the best of the incremental run. This result is suggestive, but we do not have enough data to be able to report anything statistically significant about the advantages of doing incremental evolution at this low-level of task.

6.2.3 Moving Target

Following a moving target can be thought of as a generalised version of static target approaching. Hence we tested a number of the evolved small target locators with a white cylinder (of similar width) substituted for the target; this was pushed around the gantry area in a series of smooth movements. The tracking behaviour of the control system that generated the behaviour shown in Figure 7 is illustrated in Figures 8 and 9. To understand how this was achieved, we analysed it.

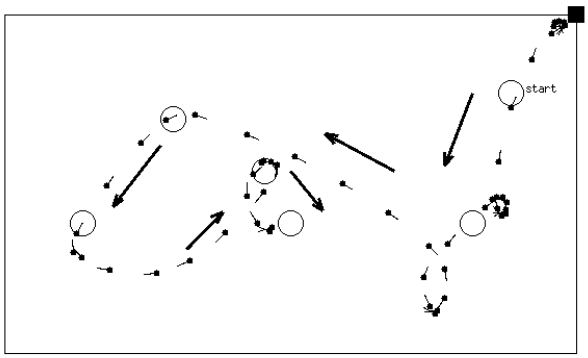

Figure 9: *Further tracking behaviour of the control system that generated the behaviour shown in previous Figure.*

7 Control System Analysis

In [4] it is shown in detail how evolved control systems of the type developed here can be analysed in terms of network dynamics and the way in which the visual morphology couples the control system with the environment. It was shown how the active part of the network can be characterised in terms of major feedback loops and visual pathways. The active part of the network that generated the behaviours shown in Figures 7, 8 and 9 is shown in Figure 10, and its coupled (evolved) visual morphology is shown in Figure 12. On analysis it was seen that in this control system only receptive fields 1 and 2 are involved in generating visually guided behaviours.

Only a brief description can be given here of the workings of the network. Due to the same software error mentioned earlier in relation to the tactile sensors, unit 5 (one of the tactile input units) acts as a source of noise over the range [0,0.25]. This unintended property of the unit has been exploited by evolution to produce a tightly self-regulating system. Unit 5 feeds into the two coupled

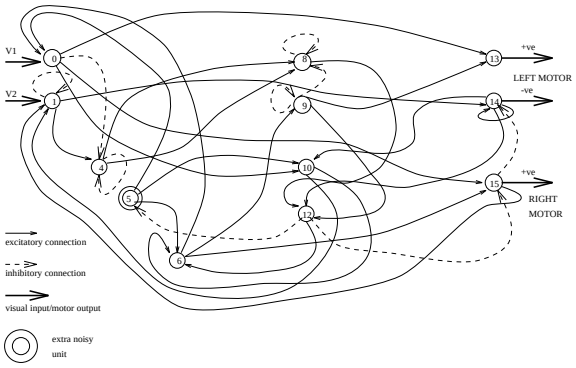

Figure 10: *Active network of the best tracker. V1 and V2 are visual inputs from receptive fields 1 and 2.*

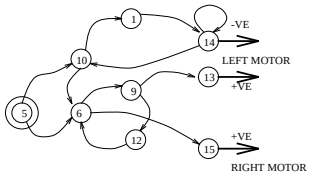

Figure 11: *Subnetwork responsible for rotations in absence of visual input.*

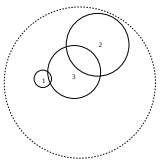

Figure 12: *The large dotted circle indicates the extent of the entire visual field available via camera and mirror. The smaller circles indicate the relative positions and sizes of the genetically specified visual receptive fields (no. 3 is not used).*

feedback loops shown in Figure 11. It can be shown that the resulting subnetwork is responsible for generating a noisy turn on the spot behaviour when visual inputs to receptive fields 1 and 2 (*v1* and *v2*) are both low (the robot is facing a dark object). When *v1* is low and *v2* is very high, unit 1 self-inhibits and the same rotational behaviour follows. When *v1* is low and *v2* is medium high the robot rotates in a medium radius circle. When *v1* is high a straight line motion follows. Due to inhibition between motor signals this straight line motion is maintained as long as *v1* remains high, irrespective of *v2*. The basic behaviour generated then, is to rotate until the white target is within receptive field 1, and then to move in a straight line as long as the target remains within the field. If the target is lost, the robot rotates until the target is again within the field of receptor 1 and straight line motion is resumed.

Further, it can be shown that in the task it was evolved to perform (small target location) this system's particular visual morphology (especially the position of field 2) was able to exploit various other visual features in the environment to ensure a rapid fixation on the target. The behaviour generated when *v2* was medium high and *v1* was low was particularly important in providing the system with surprisingly smooth tracking abilities with the moving target. Other systems tested had evolved to be too fragilely adapted to the particular task they were evaluated for; they made a lot of use of visual features other than the intended target. Consequently, when started from different positions, or with the moving target, they tended to chase reflected light spots on the walls! Clearly, great care must be taken in setting up the tasks and environments in order to get behaviour of the required robustness and generality. At the same time, it was encouraging to find a number of instances of evolved control systems that were far more general and robust than might have been expected from the evaluation function used.

8 Rectangles and Triangles

A further experiment will be very briefly described here. Two white paper targets were fixed to one of the gantry walls; one was a rectangle 21cm wide and 29.5cm high, the other was an isosceles triangle 21cm wide at the base and 29.5cm high to the apex. The robot was started at four positions and orientations near the opposite wall such that it was not biased towards either of the two targets. The evaluation function $\mathcal{E}_3$, to be maximised, was:

$$\mathcal{E}_3 = \sum_{i=1}^{i=20} [\beta(D_{1_i} - d_{1_i}) - \sigma(D_{2_i}, d_{2_i})] \qquad (4)$$

where D_1 is the distance of target-1 (in this case the triangle) from the gantry origin; d_1 is the distance of the robot from target-1; and D_2 and d_2 are the corresponding distances for target-2 (in this case the rectangle). These are sampled at regular intervals, as before. The value of β is $(D_1 - d_1)$ unless d_1 is less than some threshold, in which case it is $3 \times (D_1 - d_1)$. The value of σ (a penalty function) is zero unless d_2 is less than the same threshold, in which case it is $I - (D_2 - d_2)$, where I is the distance between the targets; I is more than double the threshold distance. High fitnesses are achieved for approaching the triangle but ignoring the rectangle. It was hoped that this experiment might demonstrate the efficacy of concurrently evolving the visual sampling morphology along with the control networks.

After about 15 generations of a run using as an initial population the last generation of the incremental small target experiment, fit individuals emerged capable of approaching the triangle, but not the rectangle, from each of the four widely spaced starting positions and orientations. The behaviour generated by the fittest of these control systems is shown in Figure 13. When started

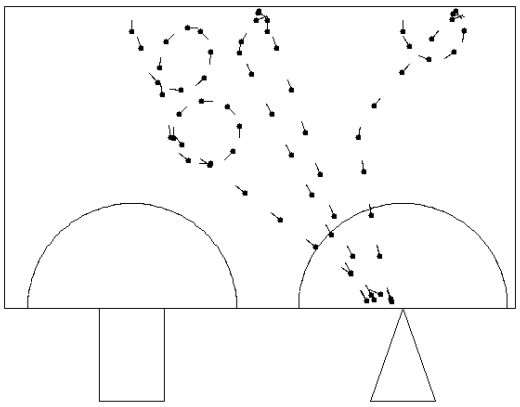

Figure 13: *Behaviour of a fit individual in the two target environment. The rectangle and triangle indicate the positions of the targets. The semi circles mark the 'penalty' (near rectangle) and 'bonus score' (near triangle) zones associated with the fitness function. In these 4 runs the robot was started directly facing each of the two target, and twice from a position midway between the two targets; once facing into the wall and once facing out.*

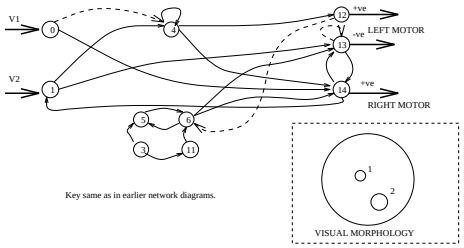

Figure 14: *Active part of the control system that generated fit behaviour for the rectangle and triangle experiment. Visual morphology shown inset.*

from many different positions and orientations near the far wall, this controller repeatedly exhibited very similar behaviours to those shown.

The active part of the evolved network that generated this behaviour is shown in Figure 14. The evolved visual morphology for this control system is shown inset. Only receptive fields 1 and 2 were used by the controller.

Whereas the fit control systems for the previous experiments only made use of one visual receptive field at a time, this one used two simultaneously. The visual morphology/networks evolved such that robots rotated on the spot when both visual inputs were low (this is effected by the subnetwork made from nodes 3, 5, 6 and 11). When the signal from receptive field 1 (v_1) is high but that from receptive field 2 (v_2) is low, the connection from unit 0 to unit 14 generates a rotational movement. When v_1 and v_2 are both medium high, the inputs from unit 1 to units 12 and 13 tend to cancel each other out whereas unit 14 is strongly activated, again resulting in a rotational movement. When v_1 and v_2 are both high, the inhibitory links from unit 0 to unit 4, and from unit 13 to itself, come into play. This just leaves unit 14 active and rotation follows. If v_1 is high but v_2 low, similar behaviour ensues. However, if v_2 is high and v_1 is low, units 12 and 14 (via 4) are active and unit 13 is inhibited. Hence straight line motion is produced. The active receptive fields were so arranged to result in the robot tending to accurately fixate on the the triangle and moving in a straight line towards it. It would often fixate on the edge of the rectangle but as it moved towards it both visual signals would go high, resulting in a rotation towards the triangle. As the robot moved towards the triangle with only v_2 high, the looming target would cause v_1 to go high. However, the (evolved) layout of the receptive fields relative to the geometry of the triangle meant that the ensuing rotational movement very rapidly sent v_1 low while v_2 remained high, and the robot carried on moving towards the triangle usually only slightly deflected from its original path. When it reached the target, depending on its orientation, the robot either stopped or slowly rotated away from the triangle and then looped back towards it. These results illustrate that tasks such as these can be achieved with extremely minimal vision systems and very small networks.

9 An Initially Converged Population

Whereas Genetic Algorithms (GAs) are normally used to search high-dimensional spaces, the modified form of GAs, 'SAGA' [2], employed here uses a genetically largely converged population, and in effect searches a relatively local space of adaptations to the current population; artificial evolution is treated as exploration, driven largely by mutation, rather than search. The population is maintained at some fairly high degree of convergence by the balance between mutation and selection. For a continuing sequence of experiments, with tasks of added complexity, the starting point in each case is the population that succeeded before, but there are different choices for the very first population.

One could start with a randomly generated population (i.e. their genotypes are randomly generated from valid symbols), which would be the normal GA technique. But often in a normal GA problem, different parts of the genotype contribute semi-independently to the evaluation function, and through the Schema Theorem [3] progress of some sort can be made from such a random start. In our case, however, the genotypes describe control systems which in turn generate behaviour, with no simple correlation between the genotypes and the behaviour; which means that, at least with encodings like the one used here, two different genotypes which both

produce promising behaviour will, on recombination, almost always produce a genotype with near-average performance — i.e. useless performance. It is only once the population has largely converged — as advocated with SAGA [2] — that recombination is likely to be useful.

For this reason, from a start with a randomly generated population, the early stages will do no more than allow some early promising candidate to dominate the population. In which case we can speed up the process, and help give some desired initial direction, by ourselves observing the first random population, choosing by eye the most promising, and seeding the next generation with clones of this one. Thereafter the population settles down to its asymptotic degree of genetic convergence from above, rather than from below. For the experiments reported here, an initial randomly generated population of size 30 was judged by eye on the intuitive criterion of 'interesting' behaviour. Two members displayed forward-moving behaviour, which altered in character when the white target was within view of the visual system, and one of these two was selected. The informal criterion of 'interestingness' allowed a clear choice, whereas the 'official' evaluation function used thereafter did not give clear preferences on this initial random population, as the scores it gave there were dominated by noise. This use of different evaluations over time is completely consonant with the underlying philosophy of this approach, that of human-directed evolution of the robots.

As has already been mentioned, the successes we have had with initially converged populations are from too small a sample of experiments to have any statistical significance. It should also be noted that the genetic encoding scheme plays an important role in determining how effective crossover is in early generations.

10 Future Work

Encouraged by the initial results with the gantry apparatus we intend to start using it in more complex experiments. In these we intend to use networks with much richer intrinsic dynamics, and more sophisticated genotype to phenotype developmental processes allowing a less restricted open-ended evolutionary process. We will explore behaviours in cluttered and dynamic environments and under changing lighting conditions.

Evaluations with the gantry using a real optic array take less than one order of magnitude longer than the early simulations we did using ray-tracing in a very simple environment. But whereas ray-tracing simulations rapidly scale up in computational requirements as the environment is made more complex, with the gantry there is no such constraint.

11 Conclusions

This paper has described a specialised piece of visuo-robotic equipment allowing us to evolve visually guided agents in the real world. It has reported on work that has demonstrated that our methods developed using simulation experiments have transferred to the real world. We have been able to evolve robust visually guided behaviours with very small populations in very few generations, even though the visual signals in the real world are far more noisy than in our simulations; this is in contrast to the difficulties experienced by others using evolutionary techniques, but with different control system building blocks [5]. A number of our evolved control systems showed interesting levels of adaptation when tested on generalised versions of the task they were evolved for, even though they use only one or two visual receptive fields and a very small network. We have demonstrated the efficacy of concurrently evolving the visual morphology along with the control networks. We find it promising that we have obtained interesting results with a simple type of network and an unsophisticated genetic encoding. Particularly since we regard both of these as being among the least powerful of the classes of networks and genetic encodings we advocate.

Acknowledgements

We thank Tony Simpson, Martin Nock, Jerry Mitchell and Harry Butterworth for engineering design and construction work on the gantry. The work was funded initially by a University of Sussex research development grant, and continuing research is now funded by the UK Science and Engineering Research Council.

References

[1] D. Cliff, I. Harvey, and P. Husbands. Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):71–104, 1993.

[2] I. Harvey. Evolutionary robotics and SAGA: the case for hill crawling and tournament selection. In C. Langton, editor, *Artificial Life III, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVI*, pages 299–326. Addison Wesley, 1993.

[3] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, USA, 1975.

[4] P. Husbands, I. Harvey, and D. T. Cliff. Circle in the round: State space attractors for evolved sighted robots. *Robotics and Autonomous Systems*, forthcoming.

[5] C. Reynolds. An evolved, vision-based model of obstacle avoidance behavior. In C. Langton, editor, *Artificial Life III, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVI*. Addison Wesley., 1993.