Swarm-Based Active Vision

Pedro Santana (Pedro.Santana@di.fc.ul.pt)
LabMag, Department of Computer Science
University of Lisbon, Portugal

Luis Correia (Luis.Correia@di.fc.ul.pt)
LabMag, Department of Computer Science
University of Lisbon, Portugal

Abstract

This paper proposes a computational distributed model for active vision. In the proposed model, action selection and visual processes progressively unfold in a parallel and asynchronous way through a set of cross-modulatory signals. The visual process is modelled with a swarm of perceptual agents inhabiting the physical agent's sensorimotor space, motivated by the ant foraging metaphor. Perceptual agents, called perceptual-ants (p-ants), perform local active vision, whereas the self-organised collective behaviour maintains global spatio-temporal coherence, i.e. a social cognitive map. A by-product of the method is the ability to maintain distributed, active and sparse spatial working memories, i.e. local maps of the environment. Experimental results with a simulated robot performing a simple navigation task, show the ability of the model to perform both robustly and parsimoniously in terms of processing.

Keywords: swarm cognition; active vision; visual attention; action selection; artificial life; autonomous robots

Introduction

The way perception self-organises is a central issue in the study of cognition. Perception is at the core of the agent's understanding of its surroundings and consequently affects every processing deriving from it. In fact, the current understanding on the role of perception in cognition goes far beyond this causal relationship. More than an information process, perception is a sensorimotor process on its own right. This view considers perception not as a provider of the cognitive system, but itself a dynamical part of it. It is currently accepted that the active role of perception is to a great extent what supports its robustness and efficiency.

Embodied cognition is the field that has been primarily interested in these aspects, in particular on the role body has on cognition. Modules within the cognitive system are seen as dynamical systems that are mutually entrained, as well as with the external world through the agent's body. This paper contributes to the field with a model that has been tested in a simulated robot performing a simple navigation task. The model aims at explaining how perception and action selection can progressively unfold, in a parallel and asynchronous way, by means of cross modulatory signals.

This mutual influence, which is closely related to visual attention, is essential to deal with the tremendously high amount of data provide by vision, which is largely redundant and unfolds in complex spatio-temporal patterns. The model proposed in this paper tackles this problem by recurring to collective dynamics of simple agents inhabiting the agent's sensorimotor and decision spaces. An agent-based modelling allows describing locally coherent active perception rules. The self-organisation of these agents lead in turn to collective patterns that are both robust and efficient, implementing an emergent global active perception policy when forming a swarm.

This paper is organised as follows. First, some of the related work is presented. Then, the proposed model is described, followed by the experimental results in the simulated environment. Finally, a set of conclusions is drawn, and pointers to future work are suggested.

Related Work

Attention is known to be an essential feature of any cognitively rich agent. Focusing on the most relevant aspects of the environment, taking into account the current context, is advantageous to increase both performance and robustness. The intricate relationships it has with other structures, such as sensory-driven pathways, complex categorical reasoning, and action selection, highlights its centrality and consequently its relevance to the understanding of the dynamics involved in an embodied cognitive system. Attention ultimately results in the motion of sense organs, e.g. eyes, towards the relevant stimulus source. This is called overt attention. A faster process is the one of mentally focusing particular aspects of the sensory stimuli. This is called covert attention.

A relevant outcome of attention research (Itti & Koch, 2001; Hayhoe & Ballard, 2005; Paletta & Pinz, 2000; Rothkopf, Ballard, & Hayhoe, 2007) is the increasing evidence that both overt and covert, i.e. with and without motor actions, visual attention mechanisms share a common neural substrate. Hence, as for the overt case, covert attention may also be partially understood as a sensorimotor coordination problem. Thus, it should be fruitful to study covert attention under the embodied cognition paradigm (Pfeifer & Scheier, 1999; Pfeifer & Bongard, 2006), in which perception is conceived as a sensorimotor process.

Visual attention is in general regarded as being composed of a sensory-driven bottom-up pre-attentive component, boosted by a top-down, context aware, attentive system. The outcome of these two processes’ interplay is a saliency map that it is in turn used to guide the serial application of a task-specific target detector. This approach highlights the dichotomy, present in the current understanding, between attentive and task-specific perceptual processes. While the for-
mer is massively parallel, the latter is understood as strictly sequential. However, a single sequential process is unlikely to exist in human brain, as it is a massively parallel structure. Hence, some middle ground must be met, which we argue it can be attained through an agent-based conceptualisation. Agents locally behave as active sequential processes, and at the same time when put together with others alike, they can generate coherent collective (parallel) behaviour.

The agent-based modelling has been successfully applied in active vision studies (Scheier, Pfeifer, & Kuniyoshi, 1998; Nolfi & Marocco, 2002; Beer, 2003; Fend, Bovet, Yokoi, & Pfeifer, 2003; Balkenius, Eriksson, & Astron, 2004; Floreano, Toshifumi, Marocco, & Sauser, 2004; Nolfi, 2005; Suzuki & Floreano, 2006; Pfeifer & Bongard, 2006; Kim & Moeller, 2006; Sporns & Lungarella, 2006; Croon & Postma, 2007; Choe, Yang, & Misra, 2008). In line with the ideas of active vision (Bajcsy, 1988; Aloimonos, Weiss, & Bandyopadhyay, 1988; Ballard, 1991), the basic principle of embodied cognition is sensorimotor coordination. With sensorimotor coordination agents are able to evolve (Floreano, Durr, & Mattiussi, 2008) in order to select the dynamical flow of sensory patterns that better suit the task, to increase the signal-to-noise ratio, to reduce ambiguities, and also to produce sensorimotor patterns that are themselves descriptors of object categories.

These studies reveal the power of an agent-based abstraction of perceptual processes. As aforementioned, single agents could only poorly mimic the brain function, which is notoriously distributed. This fact has triggered the interest on the use of swarms of perceptual agents to build distributed cognitive maps (Chialvo & Millonas, 1995). More recently, the swarm approach has been considerably employed in several robotics and computer vision problems, such as detecting face components (Mobahi, Ahmadabadi, & Araabi, 2006), roads (Broggi & Cattani, 2006), objects (Owechko & Medasani, 2005) and to perform 3-D range image segmentation (Mazouzi, Guessoum, Michel, & Batouche, 2007).

These solutions are mostly stand-alone engineered distributed perceptual systems. Under the scope of embodied cognition, they must be seen in an holistic framework, where other neural processes and agent’s body are considered altogether. Sensor planning is a relatively stable field in the active vision community that is also striving to solve this problem. In short, the idea is to plan the next action to be performed on the sensor, e.g. controlling a pan-tilt camera, that minimises the uncertainty of any internal states affecting the decision process (Nabbe & Hebert, 2003; Kwok & Fox, 2004; Patel, Macklem, Thrun, & Montemerlo, 2005; Baganov, Bimbo, Nunziati, & Pernici, 2006; Hernandez, Cabrera, Naranjo, Dominguez, & Isern, 2007; Sprague, Ballard, & Robinson, 2007). However, as these works are about overt attention, the problem of controlling covert attention during the unfolding of the action selection process, has not been considered so far.

From this survey, it is possible to conclude that it is still lacking, in the state-of-the-art, a model describing perception as a distributed process able to operate in synergy with other cognitive modules, such as action selection processes.

Rationale

This section describes the basic ideas of the proposed model. First the way perception and action selection interact will be described. Then, particular focus is given to the distributed aspect of the perceptual component.

Perception-Action-Perception Interaction

The computational cost of vision by far surpasses the cost of action selection for safe navigation (see below). This sensory bottleneck is thus one of the weak points of situated physical agents. The following describes a method to exploit the faster dynamics of action selection so as to guide the slower perceptual process. The hypothesis being tested is that the cross modulation of both action and visual processes fosters robust behaviour and cheap design (Pfeifer & Scheier, 1999). Cheap design in this context refers to the parsimonious use of the available processing resources.

Fig. 1 illustrates the proposed model, which assumes the existence of two loosely coupled asynchronous processes embedded in the agent’s control system. One of them, the action selection process, is responsible for assessing the utility of each possible motor action, by taking into account the agent’s goal and the current awareness on the surroundings’ configuration. Its representation of the environment’s configuration is asynchronously updated by the second process, the perceptual one, as the latter detects objects of interest in the environment. The assessment of the possible motor actions’ utilities is reiterated as new perceptual information arrives to the process. Periodically, or when no significant changes occur, the motor action with higher utility is selected and actually sent to the agent’s actuators. That is, the action selection process reassesses the actions’ utility several times before the final selection is taken. This is possible when action selection is a simple feedforward mechanism, whose timescale is by far the smallest in the whole system.

As described, the perceptual process modulates the action selection one. However, the symmetric interaction is also a

![Figure 1: The proposed model. Arrows represent asynchronous interactions among and within each process.](image)
useful one, as it configures an active vision process. Rather then performing a blind analysis of the sensory data, the perceptual process uses the action selection modulating signal, which is composed by the current motor actions’ utilities, to prioritise the search. This way, the perceptual process is able to detect high priority objects sooner and consequently update, in a timely way, the action selection process with relevant information. Relevance here is defined in terms of proneness to induce a change on the receiver. In other words, a detected object is the more relevant the more it induces the action selection process to re-rank high utility motor actions.

Under this perspective, perception is an active process whose operation is judiciously constrained to push action selection forward. Put in other way, the goal of the active perception system is to focus resources on those aspects of the environment where a detected object is more prone to affect the course of action selection.

Let us give an illustrative example. In the case of obstacle avoidance, perception should focus on those regions that would be reached out by the agent if the motor actions with higher utility would be actually sent to the actuators. That is, if the agent’s goal is to move forward and no obstacle has been found yet, then the action selection process would award the forward motion as the one with highest utility. As a consequence, the perceptual process would focus its resources on detecting obstacles in the region corresponding to the forward direction. If an obstacle was meanwhile detected, it would be reported to the action selection process, which would in turn reduce the utility of moving forward, and increase the utility of turning to both right and left. This decision would in turn compel the perceptual process to reduce the analysis of the region in front of the agent and start analysing more strongly its sides. If a new obstacle was then detected on the right side of the agent, it would influence the action selection to reduce the utility of turning to that side, which in turn would induce perception to focus its resources mostly on the agent’s left side. If instead, no object was found in the agent’s front region, the action selection would select the straight ahead motion, and as a result the perceptual process would also keep focusing that region, thus ignoring the potential presence of obstacles in the agent’s sides. Detecting them would be unnecessary as they would never affect the decision of moving forward, which was the initial goal altogether.

From these considerations follows that three loops co-exist in the model. One is related to the cyclic behaviour of each process, and it is represented by the self-links in Fig. 1. The time scale of this loop is a function of the complexity of the corresponding process. Objects harder to detect are associated to more complex perceptual processes, and consequently to longer time scales. The same trend is observed in the action selection process when the look ahead is increased. The second loop represents the way motor actions affect perception, and therefore closes itself through the world. The timescales involved in this case are usually rather slow, depending mostly on the agent’s dynamical characteristics. Finally, the third loop represents the loosely coupling between both action selection and perceptual processes, enabling their cross modulation. The time scale of this loop is of the order of magnitude as both action selection and visual processes.

Considerations on Parallelism

Focusing most perceptual resources on the analysis of the environment’s region associated to the action with higher utility is important, as objects detected there have the higher chances of affecting the action selection process. However, objects detected on other regions may also change the utility of the most relevant region. For instance, an obstacle dangerously close to the agent may affect all possible motion directions. Hence, perceptual resources must also be partially allocated to other regions but the one with highest utility.

The need for allocating perceptual resources to more than one region at a time raises the problem of devising a policy able to trade-off between coverage and accuracy. That is, if the system focuses too hard on the best action, objects are detected, but then little time is available to analyse the second best, and so on. A coarser analysis, that would allow a comfortable trade-off of both actions may in turn reduce the chances of detecting small objects. A similar problem is related to the fact that the closer the object is from the agent the more important it is to detect it, and so the search should be denser in the agent’s vicinities. But again it is not clear how much resources should be spent on the vicinities before start analysing the periphery.

This discussion is a reflection of our tendency to model phenomena as serial processes operating in optimal regimes. However, such optimal solution only exists if a particular agent-environment interaction is considered. In embodied systems, where the environment is dynamic and the outcome of the agent’s actions is to a large extent uncertain, such particularised optimal design will most surely result in brittle behaviour. Under the active vision perspective advocated in the previous section, both perceptual and action processes should be deeply and dynamically intricate. This internal dynamics adds to the agent-environment one, resulting in a global dynamics too complex to be tackled with case-based optimal solutions. Instead, a sub-optimal parallel analysis of the visual input is able to exhibit higher levels of robustness, as results confirm. In addition, parallelism facilitates the achievement of a proper trade-off between coverage and accuracy.

This approach also eases the regulation of the processing load across frames. A purely greedy policy would analyse the input image only enough to guarantee the stabilisation of the action selection process. This means that if there was no detected object affecting the motor action with highest utility, no further analysis would be required. Hence, only the image region associated to that motor action would be analysed. However, the second best motor action, and others, are also likely to become the best ones as the environment evolves in the agent’s field of view. The more data has been gathered in the spatial map regarding the most relevant actions in previous time steps, and not only concerning the best one, the
faster the decision can be made when the context changes. Thus, allocating resources proportionally to the utility of the several motor actions, i.e. in a parallel way, embeds the system with predictive power.

In static contexts, where p-ants are able to rapidly re-detect objects as a new image is fed, the system’s load to maintain a proper representation of the environment is low. This allows the distributed perceptual system to spend additional time analysing regions of the visual field that are not necessarily the most relevant ones. As a result, a larger region of the environment is analysed, providing the action selection process with potentially useful information to assess second best motor actions.

As the context changes, e.g. a new task is defined or the environment changes in the agent’s field of view, the second best motor action may end up as the best one, and thus requiring a corresponding focus of attention. Had the sub-optimal parallel search not been performed while the context was static, and now the system would be overloaded with the need of stabilising both action selection and distributed perceptual system. Thus, the parallel and apparently sub-optimal analysis exhibits some predictive power and also enables a robust balance of the load.

### Distributed Perception

The difficulties discussed in the previous section are in fact artifacts emerging from the attempt of, on the one hand serialising a problem that is naturally parallel, and on the other hand seeking for an optimal solution for a problem that calls mostly for robustness. A remarkable metaphor encompassing these characteristics is the foraging behaviour of army ants, which are able to leave their nests and cover large areas to search for food items in a purely distributed way. The robustness of their behaviour is evidenced by the qualitatively different patterns of pheromone trails they generate in response to different food spatial distributions (Deneubourg, Goss, Franks, & Pasteels, 1989). These ants exhibit a sort of collective intelligence (Franks, 1989), allowing the group to be seen as an individual decision making process. It is even possible to draw a mapping between insect-based swarms and neuron-based brains of vertebrate (Passino, Seeley, & Visscher, 2008).

The specific mapping between ant foraging concepts and the ones required by a robust visual process for object detection is also feasible. We propose the idea of modelling the dynamical neural activity present in the visual cortex, as the complex patterns of a collective behaviour resulting from the interaction of local active processes, i.e. what we call “perceptual-ants”, inhabiting the agent’s sensorimotor space.

A perceptual-ant (p-ant) represents a covert focus of attention to a given portion of the visual input, to which a task-specific detector is applied. Analogously, a natural ant can be seen as a mobile sensor that locally senses the environment for the presence of food items. Natural ants reinforce pheromone trails leading to fruitful locations so as to promote the rate at which food items are returned to the nest. P-ants also use a form of pheromone to recruit other p-ants to the location of detected objects. As for the natural case, too high pheromone concentration repels p-ants. This guarantees an adequate number of p-ants to aggregate around a given object. This resembles the cell synchronisation process, which is known to be associated to visual grouping in the visual cortex (Singer, 1999; Yazdanbakhsh & Grossberg, 2004).

Patterns in human brain, may break when the input visual input changes, but rapidly re-emerge in order to track the dynamics of the environment. Ant colonies are also known to be endowed with similar resilience in what regards tracking environmental changes. Thus, this ability of ant colonies can be used as metaphor to model tracking behaviour in distributed visual processes. Tracking is about the ability of updating the estimate of a given object’s position in the visual field, which can change for instance due to motion. This in general requires the tracking entity to perform a local search with the goal of re-detecting the object. Accordingly, if a p-ant determines that it has lost the object it has been focused on, its tracking behaviour will initiate a local search. As their natural counterpart, p-ants also exploit other p-ants pheromones to boost this search. In addition, p-ants are also sensitive to the physical agent’s motion, allowing them to perform the respective compensation.

This ability of tracking and motion compensation endow the visual system with an intrinsic distributed and active spatial map of the environment. That is, each object is represented by a set of p-ants that actively seek to maintain lock.

It is largely recognised that the amplification of random fluctuations is essential to allow the emergence of novel situations in foraging tasks, and in self-organised systems in general (Bonabeau, Dorigo, & Theraulaz, 1999). In this line, p-ants behaviour must also encompass a probabilistic component. We add the following advantages of using noise in search tasks. The goal of explicitly using random fluctuations allows p-ants to actively search the surroundings with complex search patterns, without their explicit coding. Moreover, the use of stochastic behaviour has the additional advantage of naturally handling noisy data. A deterministic program would require a, still brittle, extensive exception handling facility. In short, random fluctuations are essential to foster both cheap and robust design. In fact, neural variability to the same stimulus has supporting evidence of being able to explain to some extent Bayes-optimal human brain performance, with simple neural circuitry (Ma, Beck, Latham, & Pouget, 2006).

### Proposed Model Detailed

#### Experimental Setup

In this study a wheeled robot, simulated in Player/Stage (Gerkey, Vaughan, & Howard, 2003) (see Fig. 2), with the goal of simple navigation is used as test-bed. Simple navigation requires obstacle detection (visual process) and obstacle avoidance (action selection process). Since the simulator does not provide range sensing through vision, for instance using a stereo head, a laser scanner had to be utilised instead. The laser scanner measures ranges up to 8m along a plane
parallel to the ground, and it has a field-of-view of 180°.

Figure 2: Experimental setup. The shaded area corresponds to the original laser scanner field-of-view (180°). Obstacles are represented as black lines and dots.

Since this paper is about vision, it is necessary to emulate what the robot would see with a camera using the range data provided by the laser scanner. For this purpose, the obstacles detected with the laser scanner are directly drawn in the synthetic image as black disks (see Fig. 3). The outcome is a 300 × 150 input image representing a bird-eyes view of the environment. With this configuration, a black pixel corresponds to an obstacle, and its position in the image is directly proportional to its position in the robot’s workspace.

Figure 3: Synthetic visual input generation. The image on the left is a schematic bird-eyes view of the robot in a given environment. The robot is the circle at the bottom and it is pointing forward. The dashed straight lines correspond to the laser beam at a given angle. The circle and the square at the top are obstacles. The image on the right corresponds to the synthetic input image, generated from the laser readouts.

Note that real images acquired from a vision sensor have perspective, i.e. the distance between any two points in the world is not proportional to the distance between the projections of those two points in the image plane. The fact that this not occurs in the generated synthetic input image is not limiting the generality of the results. With some projective geometry, and a set of realistic assumptions, it is possible to transform real images on a bird-eyes format.

As said, the goal of the distributed perceptual system is to search for objects, in this case obstacles, in the input image. That is, p-ants move in the image space and apply a specific detector, i.e. perform an obstacle presence test, in their position. In this case the detector is as simple as checking whether the pixel where the p-ant is on, is black. In natural images the detector would be necessarily much more complex (see Fig. 4) and thus time consuming. The proposed model’s performance will be evaluated in part with respect to the number of tests performed by the swarm, and not the timings of their individual elements. Thus, despite its simplicity, this setup effectively allows the assessment of both performance and robustness of the proposed model.

Figure 4: Obstacle detection in natural images. Detecting obstacles in natural images is about determining whether the region surrounding a given pixel, here represented by a square, corresponds to an obstacle. It may even be necessary to classify the pixel either as “tree”, “ground”, “rock” or “vegetation”. This usually requires the time consuming computation of large filter banks.

Overall System Architecture

The proposed model is distributed and asynchronous, fostering its plausibility for parallel architectures, such as the brain. However, in order to test it in current computer technology without the considerable overhead introduced by the use of threads, it is necessary to serialise its execution. Other parallel models, such as cellular automata are also typically implemented as serial processes. For this purpose it is necessary to allow each component of the system to propagate, one at a time, its state for one unit of time. The use of a synchronous update policy, i.e. when components are updated according to a given sequence, has been shown to generate results different from those obtained with an asynchronous (parallel) one. For the context of the current work this aspect is neglected, under the assumption that the nature of the proposed system is not as coupled as cellular automata. Nevertheless, future work will address this problem by implementing full parallelism through intermediate asynchronism (Grilo & Correia, 2007), which is implemented with probabilistic updates.

The pseudo-code of the current system’s implementation is sketched in Algorithm 1. The algorithm has two nested loops,
being the inner one responsible for updating both action and perceptual processes for a given number of iterations, or until the best action selected stabilises for a given number of iterations. In this loop, the action selection process updates its action utility vector (Line 12) taking into account the obstacles already reported by the perceptual process, and selects the currently best action accordingly (Line 13). Then, p-ants that are analysing the image are one-step iterated one at a time (Line 18). That is, p-ants select the next position and move towards it. In this step, whenever a p-ant detects an obstacle reports it to the action selection process so that in the next iteration the latter can update the action utility vector (Line 12). All p-ants that die, due to moving out of boundaries or because they are too old, are removed from the system (Line 19). Finally, the swarm of p-ants analysing the image is repopulated by recruiting new p-ants according to the utility vector and current agent’s speed (Line 20).

**Algorithm 1 Overall system’s execution pseudo-code**

Require: desired_heading, desired_speed

1: previous_best_action ← 0
2: while true do
3:   perception::wait_for_new_image()
4:   motion ← get_motion_estimate()
5:   perception::reset_pheromones()
6:   action::reset_obstacles_set()
7:   perception::compensate_motion(motion)
8:   perception::iterate()
9:   iterations ← 0
10: while (iterations < 20) and (stable < 10) do
11:     iterations ← iterations + 1
12:     utility ← action::utility_update(desired_heading)
13:     best_action ← action::get_best_action(utility)
14:     if (best_action = previous_best_action) then
15:         stable ← stable + 1
16:     end if
17:     previous_best_action = best_action
18:     perception::iterate()
19:     perception::remove_corps()
20:     perception::recruit(motion, utility)
21: end while
22: action::send_to_actuators(best_action, desired_speed)
23: end while

The inner loop operates on a single input image, which is refreshed by the outer loop. The outer loop starts by waiting for the availability of a laser scanner (Line 3), which is immediately used to create the synthetic image. New laser readouts are available at approximately 10 Hz. Afterwards, any pheromones deployed by the swarm of p-ants is reset (Line 5), as well as the obstacles set in the action selection process (Line 6). Note that p-ants themselves are not reset or removed between images. They are only removed when they die in the inner loop (Line 19). In fact, the longevity of p-ants allows them to survive across several images. As they survive old images, they must be compensated by the robot’s motion in order to maintain their position in the world frame between images (Line 7). Then, the perceptual process is asked to iterate in order to inform the action selection process of the obstacles where p-ants are currently on. Remember that the obstacle set has been previously reset (Line 6). Finally, the inner loop is executed (Line 10-21) followed by the actual execution of the best action (Line 22).

The following sections describe these steps in more detail.

**Action Selection Process**

The action selection process is assumed to produce a multi-valued output at each time step $n$, i.e. it generates a vector $u_n = (u_n^1, u_n^2, \ldots, u_n^k)$, where $u_n^k \in [-1, 1]$ is the utility of performing action $k \in K$. Actions for the purpose of this study are instantiated as linear trajectories that can be performed by the robot. The assessment of each action’s utility is performed taking into account the registered objects set, $O$, which grows as detected objects, i.e. obstacles, are reported by the distributed visual process. The reporting of detected objects is the way the visual process feeds back to the action selection process.

**The computation of each linear trajectory’s utility** (Line 12) is based on a method for fast obstacle avoidance (Santana & Correia, 2006). The goal is to determine which linear trajectory $a^k_n$ the agent should take to produce faster progress in the direction of the goal heading $h$ (relative to the world frame) without hitting any obstacle in set $O$. Fig. 5 illustrates the set of $k = 81$ possible linear trajectories relative to the robot’s main axis of motion, as well as their utilities for a given example.

First, the obstacles in $O$ are used to determine, in the configuration space, the maximum distance, $d^i_n$, the robot is able to travel along each possible linear trajectory, $k \in K$, without hitting any obstacle. Then, gaps in the environment are searched in order to assess the free space connectivity between consecutive linear paths. Theoretically a gap is a discontinuity greater than the width, $w$, of the robot in the $d^i_n$ feature between two adjacent linear paths, through which the robot is able to cross. From the two linear trajectories involved in the discontinuity, the one with greatest $d^i_n$ is considered to be a corridor. Additionally, all linear trajectories with $d^i_n$ equals to the sensor range are also considered corridors. The utility for all corridors is computed as follows,

$$ u_n^k = \beta_1 \cdot d_n^k \cdot \cos(|\angle - h|) - \beta_2 \cdot d_n^k \cdot \sin(|\angle - h|) $$

where $\angle$ is the angle between linear trajectory $k$ and the robot’s main axis of motion, and $h$ is desired heading $h$ transformed to the robot’s frame of reference. The utility is higher.
in corridors whose \( d_k^n \) projected on the desired direction of motion is bigger (first term) and whose lateral displacement from that direction in smaller (second term). \( \beta_1 \) and \( \beta_2 \) are empirically defined importance weights, which in this study have been set to 0.5. Linear trajectories which are not corridors have utility zero.

The robot’s linear speed is provided by the user, whereas the angular speed is proportional to the angle of the selected corridor as well as the current linear speed. This simple procedure is sufficient to provide smooth obstacle avoidance.

In sum, the action selection process computes the action utility vector according to the set of obstacles that have been asynchronously reported by the visual process. In turn, the action selection process feeds back the visual process with the action utility vector. When the action with highest utility (Line 13) stabilises for a given number of iterations, typically 10, it is used to actually generate a motor action (Line 22). In addition, the obstacles set \( O \) maintained in the action selection process is cleared every time a new image arrives to the system.

**Visual Processing**

For the obstacle avoidance case the goal of visual processing is to detect the obstacles that influence the obstacle avoidance policy. All other obstacles are irrelevant and therefore have low priority. To be this selective, the search strategy is modulated by the incoming utility vector \( u_n \), provided by the action selection process, enabling an informed focus on the aspects of the environment that are more prone to affect the action selection process. Other spatio-temporal constraints can also be defined, such as: obstacles closer to the robot have higher priority.

The following summarises the distributed perceptual process that has been devised to cope with these requirements. The process is inspired by foraging strategies. In particular, a swarm of simple homogeneous agents, i.e. p-ants, leave the nest to search for food items, which are subsequently tracked until they are fully consumed. After consumption, p-ants return to the nest (in our implementation they are just removed). While tracking food items, p-ants leave a pheromone trail that can be sensed by others to potentially help them to re-detect a lost food item. Found obstacles, i.e. food items, are registered in the action selection mechanism by the p-ant responsible for the detection. The action selection modulating signal, i.e. \( u_n \), is in turn used to modulate the recruitment of new p-ants in the nest, as well as to boost their motion towards relevant directions. As for biological systems, the system’s stress should be able to affect its responsiveness. Stress here is defined in terms of robot’s speed: the faster the robot moves, the more p-ants are recruited for the perceptual task.

**Recruitment**

The nest is located in the centre of the image’s bottom, thus compelling p-ants to start their search in the close vicinity of the robot (see Fig. 6(a)). The rate at which p-ants are recruited is a function of the robot’s speed, \( s_n \in [0,1] \). At each iteration a probabilistic test is performed, \( P(x_1) < s_n \), to check whether more p-ants can be recruited in the current iteration, where \( P(x_1) \in [0,1] \) represents samples from an uniform distribution. With this test, the system’s load matches the level of stress. In case the first test succeeds, for each possible linear trajectory within the robot’s field-of-view (90°) a second test is performed, \( P(x_2) < u_n \), where \( P(x_2) \in [0,1] \) represents samples from a uniform distribution. If this test also succeeds, then a p-ant is recruited with an initial position defined in polar coordinates in the image plane \( (\theta_n, \rho_n) = (\alpha_k, 0) \), which lies on corresponding linear trajectory \( k \). With the second test, the higher the utility of a given linear trajectory, the higher the chances of recruiting a p-ant to move on the corresponding image region. Note that this 1 : 1 mapping between robot’s linear trajectories and lines in the image plane is only possible due to the simplifying assumptions, namely the absence of perspective. Otherwise a proper transformation would be required.

Notice that, in the system there is no explicit control of the maximum number of p-ants that can be deployed. In fact, this number varies according to the environment’s configuration, and consequently it is an emergent property resulting from the immersion of the system in a specific situation.

**Searching Behaviour**

After being located for the first time in the image plane, each p-ant’s position is updated at each iteration with small random motions defined as additive Gaussian noise,

\[
(\theta_n, \rho_n) = (\theta_{n-1} + \lambda \cdot N(0,1), \rho_{n-1} + |N(0,1)|)
\]

(2)
P-ants have an age that increases at each iteration. When a p-ant achieves a top age it is said to be dead, and is consequently removed from the system.

In the detection process, p-ants apply their detector, which in this simplified case simply checks whether the pixel where the p-ant is on is black. As mentioned before, in natural images (e.g. Fig. 4) the application of a robust object detector is a rather time consuming process. Thus, in order to account for this aspect, the application of the detector should be reduced as much as possible. To attain this, before applying the actual detector, the p-ant checks whether any other p-ant has already evaluated the pixel in question. If so, then it reuses the result. This communication is done implicitly through stigmergy. P-ants applying the detector report the outcome by deploying a corresponding pheromone on a shared medium (see Fig. 6(b)), which is then locally inspected by others. This pheromone takes value 1 if an obstacle has been detected in the pixel in question, and 0 otherwise.

In case the detector returns a positive, the p-ant reports the fact to the action selection process and deploys a second type of pheromone, which ranges between $[0, 255]$, in a second shared medium (see Fig. 6(c)). This pheromone is additively deployed, with a top-view pyramidal shape, with top magnitude of 20 and linear decay outwards, spanning 40 pixels. If the random motion (first step in the search procedure) would take the p-ant to a pixel with concentration of pheromone higher than $p$, then the p-ant stands still during the current iteration. This procedure controls the number of p-ants co-habiting the same obstacle. For the purposes of this study, $p = 30$. Subsequently, it starts tracking the detected obstacle.

**Tracking Behaviour** Sensations change as the robot moves. However, these changes are seldom dramatic, thus enabling the use of the previous step state to predict the next one. Otherwise, the whole perceptual process would have to be repeated from scratch at each iteration. So, the small changes are compensated by means of tracking obstacles across frames. P-ants accomplish this by moving in order to compensate for the robot’s motion according to simple homogeneous transformations. Robot’s motion can be estimated for instance with odometry, which by being an estimate is regularly contaminated with noise. Being unable to perfectly compensate for the robot’s motion, p-ants will eventually lose track of the obstacles, hence requiring a mechanism to locally search for them, i.e. to operate in closed loop. This local search is far more efficient than searching all obstacles from scratch. A p-ant performs local search by randomly changing its position around anchor point $(\theta_n^a, \rho_n^a)$ until the obstacle is re-detected,

$$ (\theta_n, \rho_n) = (\theta_n^a, \rho_n^a) + (N(0, 1), N(0, 1)) \cdot \sigma_n $$

where $\sigma_n$ is updated with $\sigma_n = \min(\sigma_n - 1 + 1, 10)$ whenever the detection fails, allowing the search to spread more and more at each iteration, and thus raising the chances of re-detecting the obstacle. $\sigma_n$ is set to zero whenever the obstacle is (re-)detected. In the same situation, the anchor point is set to that position. The anchor point also changes to the p-ant’s position if the local pheromone is higher then the one at the current anchor’s location and lower than $p$. The pheromone here allows p-ants to help each other, through stigmergy, on re-detecting the obstacle. Since p-ants may be in tracking mode for several frames, the anchors themselves must also be compensated for the robot’s motion. See Fig. 7 for an illustrative example of the tracking behaviour.

Lasts to say that whenever the obstacle is (re-)detected, or a motion compensation is performed, the new obstacle position is registered in the action selection process.

**Lifetime and Activity Decay** P-ants have an age that increases at each iteration. When a p-ant achieves a top age it is said to be dead, and is consequently removed from the sys-

---

**Figure 6**: Graphical output generated by the visual process for situation depicted in Fig. 2. and Fig. 5.
Figure 7: Graphical output generated by the perceptual process with erroneous motion estimate to illustrate the tracking capabilities. Although the robot does not move, the motion estimate, which is noisy, reports a straight ahead motion. By making the motion compensation, the p-ants move away from the obstacle. As the time unfolds (from top to bottom) they manage to find the way back to the obstacle. Red lines connect anchor’s and p-ant’s current position.

Figure 8: Graphical output generated by the perceptual process to illustrate the mapping capabilities in two sequential situations.

Experimental Results

An experimental setup has been prepared to show the ability of the system to produce robust and efficient obstacle avoidance with reduced computational cost. All experiments were carried out on a Centrino Dual Core 2GHz running Linux/Ubuntu. Fig. 9 depicts the simulated environment where the experience took place. The environment has two distinct regions. The first one, from which the robot starts
the run, is open and contains large obstacles in the far-field. The second region, on the other hand, is cluttered with small obstacles.

Since obstacles are near the robot in the second region, a thorough sequential analysis of the image would allow their fast and accurate detection. However, in the first region, such approach would hinder a fast detection of the obstacles in the far-field, which in this case are the ones relevant for the action selection process. The process would take too long because the whole free space between robot and obstacles would have to be carefully analysed. Thus in this case coverage should be preferred to accuracy.

This test intends to demonstrate that the proposed method enables a consistent and proper handling of this difficult trade-off, i.e. coverage vs accuracy, across both regions without explicit context awareness.

Figure 9: Experimental setup in the simulated environment with an area of 16 m × 16 m. The robot starts from the bottom-left aligned with the arrow, which also represents the desired heading in the world frame. Red rectangles represent the robot’s outline along the run.

During the run the robot is asked to travel at a speed of 1.0 ms⁻¹. Position (in meters) and heading (in radians) estimates are polluted with additive Gaussian noise characterised by 0.1·N(0, 1) and 0.001·N(0, 1), respectively, where N(0, 1) represents samples from a Gaussian distribution with mean 0 and variance 1. With this error profile, the re-detection of tracked obstacles was required nearly in almost iterations. The produced path is smooth and fairly optimised. That is, in order to progress rapidly along the desired heading, the robot cuts the way every time it is possible. In terms of performance, Fig. 10(a) plots the computational cost of both action selection and visual processes. The additional cost of image acquisition and motor actuation, which together take ≈ 100 ms, has not been considered. The average of 1.7 ms is residual when compared to typical cost of vision-based obstacle detectors, ≈ 100 ms. Thus, the overhead caused by the active vision infrastructure will not be noticed when applied to real-life obstacle detectors.

In fact, the low average number of obstacle detection tests (see Fig. 10(b)) shows that the method is able to produce judicious collective dynamics, i.e. to robustly maintain an adequate coverage vs accuracy trade-off, along with a parsimonious allocation of resources. The average number of 575 is only 2.3% of the total number of pixels available in the 300 × 150 synthetic input image, constrained by a 90° field-of-view. Hence, with a parsimoniously computational cost, the proposed method is able to robustly reduce the focus of attention to only a small fraction of the robot’s field-of-view.

It is also noteworthy that the system is already performing mapping and tracking. These two embedded features are typically considered as processes independent of the detection itself. The proposed method exploits these embedded additions to opportunistically focus attention on obstacles detected in previous frames, which in turn affect immediately the action selection, whose modulatory effect results in promptly changing the focus of attention to other image regions. The robust and efficient global focus of attention emerges from these interactions, in addition to the local behaviour of each p-ant. Achieving this complex behaviour with a centralised solution would be just too complicate.

**Conclusions**

A distributed model for active vision was proposed and validated in a simulated robot, performing a simple navigation task. The proposed model allows both action selection and vi-
sual processes to progressively unfold in a parallel and asynchronous way, by being mutually affected.

The proposed model differs from previous work in active vision research on the loosely coupling aspect of both interacting processes. In connectionist models, such as the ones typically considered in embodied cognition, for instance, there is no clear distinction between vision and action components. Being so tightly coupled, those approaches are difficult to modularise and consequently to scale. Conversely, another typical approach is to decouple both processes to the point they only share information in a master-slave way, such as in typical sensor planning strategies. This forces active vision to operate at longer time-scales, which has little utility for covert active vision.

The visual search process has been modelled with a swarm of simple agents, i.e. p-ants, based on the ant-foraging metaphor. Although the use of multiple agents in image analysis is not new, it is the first report on its interaction with action selection. That is, p-ants report opportunistically their results to the action selection process at the same time they are guided by it. In short, p-ants perform local active vision, whereas the self-organised collective behaviour maintains global spatio-temporal coherence. It is also the first report on a distributed, sparse and active spatial working memory (local map), which emerges as a by-product of the swarm-based approach. This is an example of the self-organisation of a relevant component of cognition, i.e. spatial mapping, in line with the emergent systems approach to the understanding and synthesis of cognition (Vernon, 2008). Thus, in addition to the engineering usefulness of the proposed method, the conclusions derived from it may also contribute to our understanding on the self-organisation of embodied cognition in general.

The promising results of the proposed model support growing evidence of the potential of the swarm cognition approach (Passino et al., 2008). The fragmentation of the whole behaviour into simple local rules fosters robustness. The system is locally coherent and consequently robustly faces unforeseen situations.

Future work will centre mainly on the migration of the concept to a real-life application, such as stereo-based obstacle detection for all-terrain environments (Santana, Santos, Correia, & Barata, 2008). The goal of focusing on this application is twofold: it is rather complex and refers to one of the basic, and ubiquitous, sensorimotor capabilities in the animal kingdom. In addition, a new perspective for active vision has been given, where perception is seen as an active process whose operation is judiciously constrained to push action selection forward. This statement can be considered as a criterion, a meta-heuristic, whose plausibility as a driving force to the autonomous development of active vision could be explicitly modelled and experimentally tested. Finally, the proposed model can be further extended to match the ant foraging approach more closely, such as using advanced normalising operations when recruiting p-ants, so as to improve the exploration of the environment. The link between neural variability observed in the human brain and the role of fluctuations in ant foraging could also be further studied. It may be one of the organising principles explaining the emergent optimal behaviour exhibited by both systems.

Acknowledgments
We would like thank the workshop organisers and the two anonymous reviewers for their insightful comments and suggestions. This work was partially supported by FCT/MCTES grant No. SFRH/BD/27305/2006.

References


