ON THE WATCH

Tony Belpaeme and Andreas Birk
AI-lab, Vrije Universiteit Brussel
Belgium

97RO007

Draft version
Accepted at the ISATA Conference 97, Florence, Italy, 1997.

ABSTRACT
In this paper we describe the benefits of vision for autonomous vehicles in a concrete real-world set-up. The autonomous vehicles, implemented in the form of small robots, have to face two basic tasks. First, they have to do autonomous recharging. Second, they are required to do some “work” which is “paid” in energy. We present a way to let the robots solve these tasks with basic sensors. In doing so, we focus on navigation as crucial problem. Then, vision is introduced. We argue for using the active vision framework and present an implementation on our robots.

INTRODUCTION
At the VUB AI-lab we are working with several autonomous robotic vehicles in a special experimental set-up. This so-called ecosystem is inspired by biology [McFarland, 1994] and has been successfully implemented and used in Artificial Intelligence research [Steels, 1994; McFarland and Steels, 1995; Birk, 1996; Steels, 1996a; Steels, 1996b]. Apart from the basic research issues involved in this previous and ongoing research, the ecosystem includes interesting features in respect to more application-oriented robotics and especially in respect to control of autonomous vehicles.

In previous experiments the robots were equipped with bumpers, light-sensors, active infrared-sensors, and energy-sensors. Due to the recent advances in hardware, providing inexpensive and small devices with respectable computing power, vision becomes feasible for our robots. This paper deals with the first results of using vision on our robots.

The paper is structured as follows. The section “The VUB ecosystem” describes our basic set-up including some technical details of the robots. In “Navigation for autonomous refueling and working” the problem of navigation in the ecosystem is addressed. Several ways of solving this task are presented. The following section “Vision: an overview” sketches common approaches in vision. “Vision in the ecosystem” gives an introduction to the way we use vision on our robots. The sections “The charging station” and “The competitors” describe respectively how two important parts of the ecosystem are recognized with vision. The section “Other modules” deals with the perception of other robots and beneficial side effects that can be exploited in addition. “Integration into behavior system and sensor fusion” describes how vision merges into the existing design of the robots. The section “Implementation” gives some technical details. “Conclusion and future work” ends the paper.

THE VUB ECOSYSTEM
The basic ecosystem consists of small autonomous vehicles, a charging station, and competitors (figure 1). The vehicles are small LEGO-robots (figure 2) with a carried-on control-computer. This

computer consists of a main board produced by VESTA based on a MC68332 micro-controller, and a Sensor-Motor-Control-Board (SMB-II), which was developed at our lab [Vereertbrugghen, 1996]. At the moment research is underway to enhance the robot corpus by using a sandwiched skeleton and professional motors and gears provided by MAXON. The standard sensor-equipment of the robots is as follows:

– Two bumpers in the front and in the back respectively.
– Three active infrared sensors.
– Two white-light and two modulated light sensors.
– Internal voltage and current measuring.

The SMB-II features additional binary, analog, motor-control, and ultrasound interfaces; allowing an easy attachment of further sensors and effectors. Eight secondary NiHM batteries providing 1.1Ah at 9.6V power the robots.

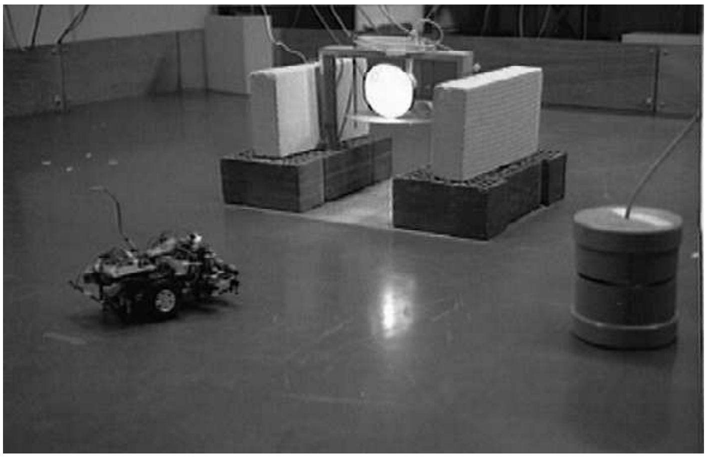

Figure 1: the ecosystem with the charging station (upper middle), a robot vehicle (bottom left), and a competitor (bottom right).

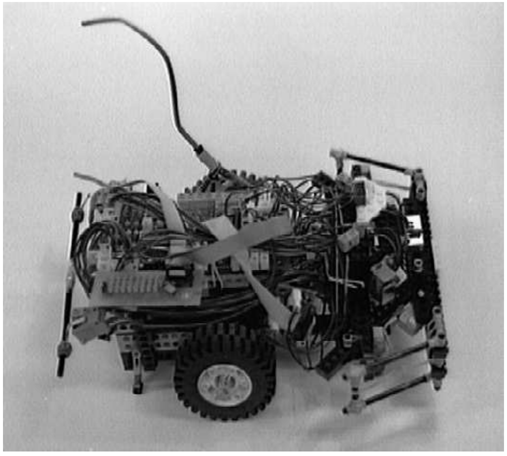

Figure 2: a robot vehicle

The robots can recharge themselves in the charging station. In doing so, there are two crucial questions involved. The actual process of recharging and the navigation problem of finding the charging station. We will ignore the first question in this paper and focus on the second one. Especially the benefit of vision for that task will be discussed in some detail later in the paper.

The competitors in the ecosystem are boxes housing lamps. They are connected to the same global energy-source as the charging station. Therefore, they consume some of the valuable resources of the robots. But if a robot “knocks” against one of these boxes the light inside the box dims. So, more energy is available for the robot in the charging station. After a while the lamps start to light again. Though this scenario is motivated by biology and designed for research on intelligence it is related to an economic viewpoint [Birk and Wiernik, 1996] as well. The “fighting the competitors” can be seen as doing a task, which is paid in a “natural” currency for robots: electrical energy. Therefore, this can be seen as a working task for the robots.

NAVIGATION FOR AUTONOMOUS REFUELING AND WORKING

As mentioned in the previous section two basic modes of the robotic vehicles can be distinguished:

1. ***refuel-mode*** including
   – navigation towards the charging station
   – staying in the charging station (picking up charge)
   – leaving the charging station (to avoid disastrous overcharge)

2. ***work-mode*** including
   – navigation towards the competitors

- “attacking” the competitors
- stopping the attack

The issues involved in the actual recharging during the refuel-mode are discussed in some detail in [Birk, 1997]. The actual “attacking” of the competitors can be achieved through control in a behavior-oriented design [Steels, 1990]. In this paradigm, the robot is not programmed in a procedural manner, but the desired performance is instead achieved through interaction with the environment. This phenomenon is denoted as “emergence” [Steels and Brooks, 1993]. We will return later in this paper to the “attacking”-behavior and discuss a concrete implementation as an example of behavior-oriented design. In the remainder of this section we will have a closer look on the options for navigation.

One possibility to navigate the vehicles is to use *dead-reckoning and a map*. Though this approach seems to be rather feasible at first glance it bears several problems. First, our robots have imprecise gearing and various other sources of error. This can be solved -to some extent- by using more elaborated -and more expensive- versions of the robots, which are underway as mentioned before. But the crucial problem is that a map has to be provided which is not static. The competitors move as the robots push them. Therefore, they do not have fixed positions. So, a human is required to constantly update the map, or the robots must have some learning capabilities. Human interaction is undesired, as we want autonomy. Learning would require at least some feedback about the position of competitors and therefore need at least one more additional “locating-mechanism”.

Another way to guide the robots is the usage of an *overhead-camera*, which overlooks the ecosystem in a bird’s view and tracks the vehicles. This approach is common in Artificial Intelligence as it resembles grid-worlds, i.e. simulations of two-dimensional environments. For example ***RoboCup***[1], the so-called Robot Soccer World Cup, follows this line. This option is technical feasible in our set-up and has been used for analysis and documentation purposes. Still, we restrain ourselves from using it for navigation for the following reasons. First, it is not “natural”. No natural being depends on or profits from a “global observer in the skies”. Second, this approach is restricted to toy settings. For example, guidance of medium or large-scale vehicles is not feasible.

Beacons are the standard way in which our robots navigate. The charging station is equipped with a bright white light and the competitors emit a modulated light signal. The robots have two sensors for each kind of signal respectively. This allows them to do simple photo-taxis: if the signal of the left sensor is stronger than the one on the right sensor, a slight right turn is imposed on the robot’s default forwarding, and vice versa. The photo-taxis towards the competitors is in a behavior-oriented design sufficient to realize the “attacking”, provided the robot is equipped with a general purpose touch-based obstacle avoidance. The robot is first led by photo-taxis towards a competitor, and bumps into it. The touch-based obstacle avoidance causes the robot to retract, the attraction of the light of the competitor causes it to advance again, and so on. The robot “knocks” as a result against the competitor until the light inside is totally dimmed. Note that the number of knocks is not programmed into the robot, but it emerges from the interactions of the robot and the competitor. Some, competitors can be “stronger”, i.e. require more knocks, than others.

Another option for navigation is *on-vehicle vision* as enhancement of the above described photo-taxis. It is discussed in detail in the remainder of this paper.

[1] RoboCup is held for the first time in August 1997 as part of the most significant conference on AI, the “International Joint Conference on Artificial Intelligence, IJCAI”, in Nagoya, Japan. It is intended to be a standard benchmark for Artificial Intelligence.

VISION: AN OVERVIEW

In classic AI two major approaches are used to tackle the vision problem: model-based vision and query-based vision. In model-based vision a robust and accurate internal model of a domain-specific world is constructed. For example, Brooks analyses static airport scenes [Brooks, 81]. But this form of explicit reasoning is not adaptive enough and lacks performance, making it less suited for real-time, real-world applications. Some systems, which do integrate dynamic aspects (for example [Koller et al., 92]) still lack adaptive and behavior-oriented aspects and do not use task-oriented processing.

Query-based vision tries to answer questions about the visual scene by running through a network of rules. This scheme has limited interactivity and is quite unwieldy in handling real-world visual data.

General-purpose architectures, which make a detailed top-down description of the world, lack –in one way or another– adaptivity and dynamics, are not task-oriented, lack interaction with the world and the symbolic representations are not grounded in perception.

The last decade, as a reaction to these approaches, a behavior-based approach to AI –and vision– emerged. In this light the active vision paradigm evolved [Ballard, 91][Blake and Yuille, 92]. Active vision is characterized by its goal-oriented design, the integration of perception and actuation, the integration of vision in a behavioral context, the use of cues and attentional mechanisms, tolerance to temporal errors, the absence of elaborate categorical representations of the 3D world, and the relying on recognition rather than reconstruction. This all makes the visual computation less expensive and allows real-time visual interaction on –relatively– cheap systems [Horswill, 93] [Riekki and Kuniyoshi, 95].

VISION IN THE ECOSYSTEM

Autonomous robots often have to rely on a limited set of sensory devices; such as tactile sensors, various light sensors and ultrasound sensors. These sensors provide a restricted amount of information, and in most cases the information is directly related to a specific situation or object which the robot can encounter in its environment; e.g. tactile sensors are only used for touch based obstacle avoidance. These non-vision-based sensors usually lack generality.

Vision however is a much richer sensor and it provides a huge amount of data, usually more than is actually needed. Visual perception can be applied in many different situations and can be used to exploit the environment in a more thorough way than other sensors can.

The robots at the VUB AI-lab are equipped with a monocular monochrome CCD-camera. To ensure a tight relation between perception and action the visual perception is real-time and is closely integrated with the behavior system of the robot. The core of the visual perception is made up of modules each handling a certain visual cue (a cue can be anything perceived by the camera, like color, horizontal edges, motion, ego-motion), the modules each rely on domain-specific knowledge. This means that the modules are specialized to a specific task and environment, which makes them much more efficient than general-purpose approaches. The (simulated) parallel working modules continuously analyze the scene in respect to their cue and pass on the result to the behavior system.

THE CHARGING STATION

The charging station has one prominent feature, its bright white light that is clearly visible in the entire ecosystem (figure 1). The visual module for recognizing the charging station uses just this light, thresholding the incoming frame does the trick. As an extra feature, the module also calculates the approximate distance to the charging station. Since the floor of the ecosystem is flat; if the charging station is farther away, it will appear higher in the image. This is important in making the choice between heading for the charging station or working some more, a non-trivial problem which depends on the battery level, the distance to the charging station, the vicinity of competitors and other robots.

THE COMPETITORS

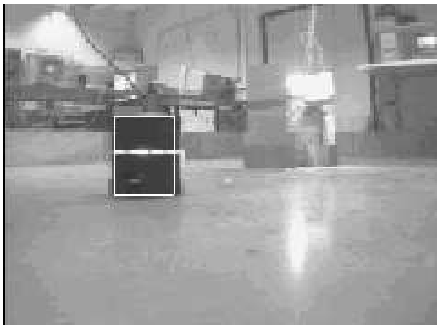

Figure 3: a competitor as seen by the robot (160×120 image). A border is placed around the competitor, meaning that it is recognised as active. To the right the charging station can be seen.

The competitors are black boxes with a lamp inside (figure 3). They are easily recognized by thresholding the image. The distance to a parasite is inversely proportional to its height and width. This allows the module to calculate a discrete (because of the discrete nature of the image) approximation of the distance to each competitor. ‘Eliminated’ competitors can be distinguished from living ones by checking the light inside the competitor, if it is on the competitor is still alive and vice versa. As result this module returns the position of closest, living competitor.

OTHER MODULES

These two modules already replicate the functionality of the light and modulated-light sensors, but some extra modules are added to aid the robot in its environment. A third module checks the ecosystem for other robots. Since the only moving objects in the ecosystem are other robots (and sometimes competitors being pushed) a straightforward way to recognize them is by looking for unusual motion in the image, apart from the ego-motion caused by the observer itself. This can be done using optical flow computation, but to save on computational resources we only use difference images to detect other moving robots. This has two drawbacks: the observer can not move during observing and the other robots have to move in order to be seen. A side effect of this module is that the observer knows when it is moving, this can be useful in situations where the robot is stuck. It occasionally happens that a robot gets stuck and it has no means to detect this (the robots are currently not equipped with wheel encoders). But if the ego-motion perceived by the camera is compared to the motor commands, it knows when it is stuck and can try to back up.

Note that the charging station and competitor modules not only are used to home in on their respective cues, they can also be used to avoid these; adding yet another way to do obstacle avoidance. The visual analysis is also quite fault tolerant: if the analysis of a few frames returns a wrong result, the robot will be corrected as soon as one good result is produced.

INTEGRATION INTO BEHAVIOR SYSTEM AND SENSOR FUSION

The common sensors used on the robots are very specific, do not give additional information on the subject they are used for (for example distance) and have a limited range. For example: the modulated light sensors have a range of roughly 1 meter, meaning that a robot can see a competitor only if it’s as close as 1 meter to the competitor. Also, recognizing more cues means adding more beacons and more sensors to the robots. Visual perception does away with all these restrictions, but this does not mean that the common sensors are superfluous. They can still be used to enrich to behaviors and can proof to be very helpful in situations where the visual perception fails. For example, when the robot is heading for the charging station, the appropriate visual module could wrongly take a reflection on the ecosystem floor as the charging station. But the light sensors do not react to reflections and the combination of both eventually work out better than the charging station module and light sensors on their own. That’s why we encourage sensor fusion: not substituting sensors with other sensors, but exploiting the interaction between perceptions to achieve new, emerging behavior.

Figure 4 shows how both visual and sensory perception can integrated into the robot’s behavior system.

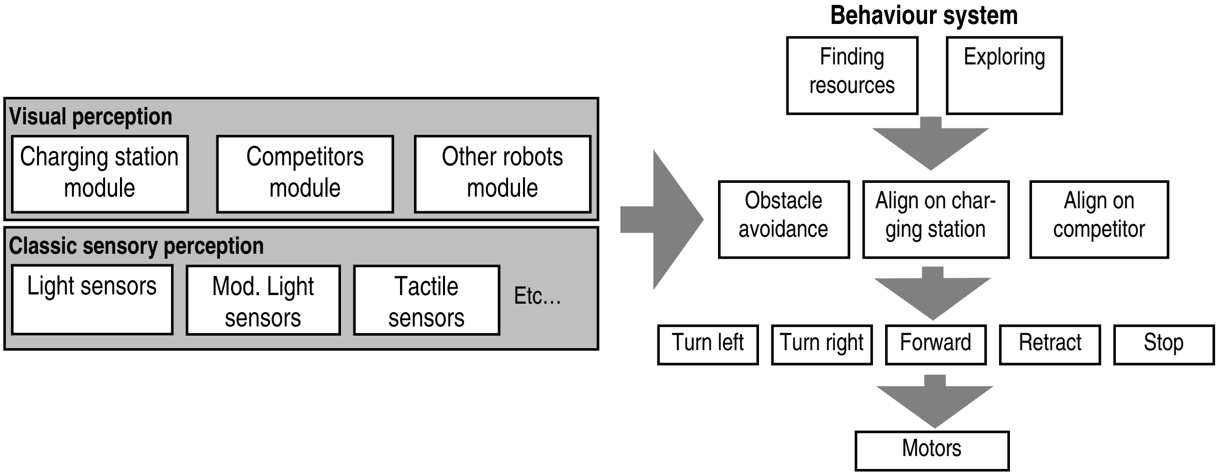

Figure 4: the behaviour-based architecture. The perceptory information (vision as well as sensors) is sent to the middle layer of the behaviour system. The behaviour system consists of three layers: a top layer, a middle layer and a lower layer (with simple modules). The actuators are the left and right motors of the robot.

IMPLEMENTATION

Active, real-time vision on the Lego-robots can be implemented in several ways. Since the analysis of visual data is computationally expensive, it can not be done by the VESTA-board carried by the robots[2]. Another solution is needed, either off-board or on-board. In the current experiments we use off-board computation. The video data is sent to a computer next to the charging station (a standard Pentium PC with a frame grabber) and the results of the analysis are communicated back to the robot. A big advantage of off-board visual computation is that during development all parameters and results can be displayed on the PC-screen. The link between the computer and the robot can be wired, using an umbilical cord, or wireless, using a video tranceiver and an asynchronous radio link for the data. This configuration gives a performance of about 5 to 7 fps, at a 160×120 resolution, which is enough for the behaviors the robot performs.

We are investigating on-board visual computation by using a Phytec TI320C50 DSP-board with a piggyback frame grabber.

CONCLUSION AND FUTURE WORK

We presented a concrete real-world set-up with autonomous vehicles in form of small robots. The robots face two basic problems: recharging and “working” in the form of attacking competitors. The advantages of using vision for these tasks were presented. In doing so, we promoted the active vision framework.

So far the actual processing of camera-data is done on a host PC. Future work includes the embedding of this processing on the robots. Furthermore, we are working on using vision on a stationary observer. This observer is a camera on a pan-tilt unit placed on the ground of the ecosystem, i.e., in the same plane as the robots. It is capable of tracking the robots and can give useful “hints”, like e.g. information on obstacles, “food”, and so on. A report on this so-called “head” is underway.

[2] Though Horswill, Yamamoto and Gavin constructed a cheap vision machine using the same processor-board (http://www.ai.mit.edu/projects/vision-machine/mobot-vision-system.html), but the processor already runs all software needed for the control of the Lego-robot and there are not enough machine cycles left for visual analysis.

ACKNOWLEDGMENTS
Thanks to Dany Vereertbrugghen and Peter Stuer for doing the design and implementation of the basic robots and ecosystem. The work of the robotic agents group at the VUB AI-lab is financed by the Belgian Federal government FKFO project on emergent functionality (NFWO contract nr. G.0014.95) and the IUAP project (nr. 20) CONSTRUCT.

REFERENCES

[Ballard, 91] Ballard, D. Animate Vision. *Artificial Intelligence*, 48 (1991), 57-86, 1991.

[Birk and Wiernik, 1996] Andreas Birk, Julie Wiernik, Behavioral AI Experiments and Economics, *Workshop Empirical AI, 12th European Conference on AI*, Budapest, 1996

[Birk, 1996] Andreas Birk, Learning to Survive, *5th European Workshop on Learning Robots*, Bari, 1996

[Birk, 1997] Andreas Birk, Autonomous Recharging of Mobile Robots, *accepted: 30th International Symposium on Automotive Technology and Automation*, 1997

[Blake and Yuille, 92] A. Blake and A. Yuille, *Active Vision*. MIT Press, Cambridge, Massachusetts, 1992.

[Brooks, 81] R. Brooks, *Model-Based Computer Vision*. UMI Research Press, Ann Arbor, Michigan, 1981.

[Horswill, 93] I. Horswill, Polly: A Vision-Based Artificial Agent. In *Proceedings AAAI-93*, Washington, 1993.

[Horswill, 96] I. Horswill, Variable binding and predicate representation in a behavior-based architecture. In *Proc. of the 4th Conf. on Simulation of Adaptive Behavior*, 1996.

[Koller et al., 92] D. Koller, K. Daniilidis, T. Thorhallson and H.-H. Nagel, Model-based Object Tracking in Traffic Scenes. In *European Conference on Computer Vision*, Genoa, Italy, 1992.

[McFarland and Steels, 1995] David McFarland, Luc Steels, *Cooperative Robots: A Case Study in Animal Robotics*, The MIT Press, Cambridge, 1995

[McFarland, 1994] David McFarland, Towards robot cooperation. In Cliff, Husbands, Arcady Meyer, and Wilson (eds.), *From animals to animats. Proc. of the Third International Conference on Simulation of Adaptive Behavior.* The MIT Press/Bradford Books, Cambridge, 1994

[Riekki and Kuniyoshi, 95] J. Riekki and Y. Kuniyoshi, Architecture for Vision-Based Purposive Behaviors. In *Proc. of the IEEE Int. Conf. on Int. Robots and Systems*, 1995.

[Steels and Brooks, 1993] Luc Steels, Rodney Brooks (eds.), *The "artificial life" route to "artificial intelligence". Building situated embodied agents.* Lawrence Earlbaum Associates, New Haven, 1993

[Steels, 1994] Luc Steels, A case study in the behavior-oriented design of autonomous agents. In Cliff, Husbands, Arcady Meyer, and Wilson (eds.), *From animals to animats. Proc. of the Third International Conference on Simulation of Adaptive Behavior.* The MIT Press/Bradford Books, Cambridge, 1994

[Steels, 1996a] Luc Steels, Discovering the competitors. *Journal of Adaptive Behavior 4(2),* 1996

[Steels, 1996b] Luc Steels, A selectionist mechanism for autonomous behavior acquisition. *Journal of Robotics and Autonomous Systems 16,* 1996

[Vereertbrugghen, 1996] Dany Vereertbrugghen, Design and Implementation of a Second Generation Sensor-Motor Control Unit for Mobile Robots, *Thesis, AI-lab,* Vrije Universiteit Brussel, 1996