

An agent based evolutionary approach to path detection for off-road vehicle guidance

Alberto Broggi *, Stefano Cattani

Dipartimento di Ingegneria dell'Informazione, Università di Parma, Parma I-43100, Italy

Available online 3 January 2006

Abstract

This paper describes an ant colony optimization approach adopted to decide on road-borders to automatically guide a vehicle developed for the DARPA Grand Challenge 2004, available from: <http://www.darpa.mil/grandchallenge>. Due to the complexity of off-road trails and different natural boundaries of the trails, lane markers detection schemes cannot work. Hence border detection is based on ant colony optimization technique. Two borders at two sides of the road (as seen by a camera fixed on the vehicle) are tracked by two agent colonies: agents' moves are inspired by the behaviors of biological ants when trying to find the shortest path from nest to food. Reinforcement learning is done by pheromone updating based on some heuristic function and by changing the heuristic balancing parameters with the experience over the last tracked results. Shadow removal has also been introduced to increase robustness.

Results on different off-road environments, as provided in the DARPA Grand Challenge 2004, have been shown in the form of correct detections, false positives and false negatives and their dependence on number of ant-agents and balancing edge-exploitation and pheromone-exploitation.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Ant colony optimization; Automatic off-road vehicle guidance; Evolutionary algorithms; Intelligent vehicles

1. Introduction

The 13th March 2004, in the desert areas of the Mojave between California and Nevada, the first competition for fully autonomous ground vehicles took place: the [DARPA Grand Challenge 2004](#). The challenge was to develop an autonomous vehicle able to complete a long and difficult off-road path from Los Angeles to Las Vegas. The course could include unsurfaced roads, sandy and rocky trails, bushes, dry lakes and, in small percentage, paved roads. The course was supposed to be about 200 miles long, was defined by several thousand GPS way-points, and was not revealed until 2 h before the event begun. The vehicles had to follow the GPS way-points as well as avoid natural and artificial obstacles, or other vehicles, without any human control.

This paper presents an artificial vision algorithm developed as a part of the TerraMax vehicle. TerraMax is the results of Oshkosh Trucks Co., Ohio State University, and University of Parma partnership ([Ozguner et al., 2004](#)). In particular the Artificial Vision and Intelligent Systems Lab of the University of Parma supplied a complete image processing system that, together with many other active sensors, performed the environment sensing: obstacle detection based on stereo vision, free space detection, and path detection based on monocular images, which is discussed this paper. [Fig. 1](#) shows the vehicle and the front cameras.

Lane detection on structured environments was already successfully faced using monochromatic monocular images by the authors ([Bertozzi and Broggi, 1998](#)) and other research teams ([Bertozzi et al., 2000](#)), based on some a priori knowledge, like the existence of lane markers. Unfortunately this approach could not work in unknown environments. [Fig. 2](#) shows some typical scenarios of the Grand Challenge path acquired by Terramax. Stones or

* Corresponding author.

E-mail addresses: broggi@ce.unipr.it (A. Broggi), cattani@ce.unipr.it (S. Cattani).



Fig. 1. The TerraMax vehicle.

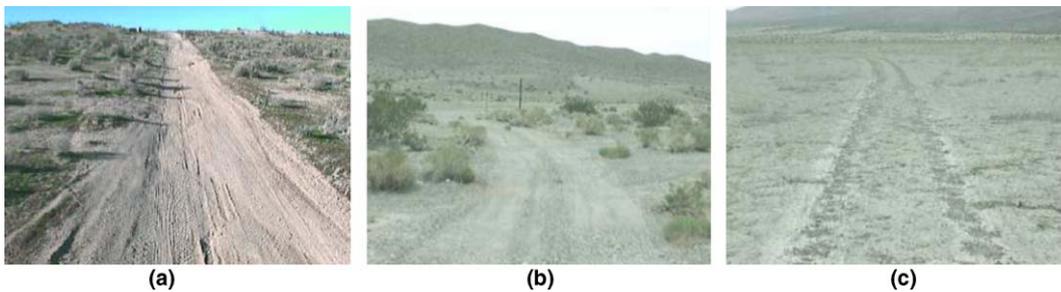


Fig. 2. Possible roads as acquired by Terramax.

bushes can be considered either road markers or can be even obstacles *inside* the road boundaries, by simply varying their position. Sometimes (Fig. 2(c)) it is just impossible, even for a human being, to localize any road.

The proposed method differs from traditional deterministic algorithm developed by other competing teams (Urmson et al., 2004), and falls into the class of evolutionary algorithms (Yim and Oh, 2003). Evolutionary methods are based on natural metaphor, and try to simulate the adaptive capabilities of the biological organism. A considerable application field is *optimization*, in which we mapped the computer vision problem of path detection. This paper is organized as follows: Section 2 will briefly introduce the evolutionary approach on which path detection is based; Section 3 will describe low-level image processing, while Section 4 will present the details of the evolutionary system. Section 5 will present some experimental results together with a quantitative analysis of the algorithm performance.

2. Evolutionary approach

The underlining idea of this work is the following: each boundary of the road is just one particular curve on the image, starting from the bottom and finishing on the hori-

zon line. If we could find some *local* properties describing the probability that a pixel belong to the boundaries of the roads, then it would be possible to formalize the problem as an optimization problem. The optimal solutions are the ones that overlap with the largest number of pixels belonging to the boundaries of the road. Clearly the cost of a solution is inversely proportional to the number of overlapping pixels.

The algorithm implemented on Terramax is based on the well known ant colony optimization (ACO) (Birattari et al., 2002; Dorigo and Stützle, 2004): a distributed meta-heuristic for combinatorial optimization problem, inspired by the communication system of the biological ants. Experimental observations (Goss et al., 1989) show that a colony of real ants, after a transitory phase, always find the shortest path from the nest to the food. This is achieved by an odorous substance, called pheromone, deposited by ants on the terrain, which attracts the following ants. Once arrived at a decision point, ants make a probabilistic choice based on the amount of pheromone they smell in correspondence to all the possible roads. Due to this *autocatalytich* process, the shortest path (the one covered in less time) will emerge as the one with the largest amount of pheromone. Therefore, returning to the vision problem, the goal became to build two colonies—one for

each side of the road (left and right)—made of agents that, moving pixel by pixel, try to find the optimal boundary curve on their side.

Ant colony optimization was already successfully applied to solve significant combinatorial optimization problem, as the Traveling Salesman Problem (Gómez et al., 2004), distributed network routing (Sim and Sun, 2003), vehicle routing (Wade and Salhi, 2004), but it is not very commonly used in computer vision; in particular there is no known application to autonomous vehicle guidance problems. The algorithm described in this paper is the only evolutionary approach to path detection seen during the DARPA Grand Challenge.

2.1. The ACO approach

A generic representation of a combinatorial optimization problems which can be solved by artificial ACO algorithms can be formalized as follows:

- A finite set X of *components* χ_i .
- A finite set A of *connections* λ_{ij} among components χ_i and χ_j in X .
- A *cost function* $\Gamma(\lambda_{ij}, t)$ with which assign to each λ_{ij} its own cost value, defined as $\Gamma_{\lambda_{ij}} \equiv \Gamma(\lambda_{ij}, t)$, where t is a time measure.
- A finite set $\Omega(X, A, t)$ of *constrains* between X and A elements.
- A set Σ of *states* defined as variable length sequences of components: $\sigma = \langle \chi_i, \chi_j, \dots, \chi_k, \dots \rangle$.
- A set $\bar{\Sigma}(t) \in \Sigma$ if *feasible states*, defined by $\Omega(X, A, t)$.
- A state σ_1 is a *feasible neighbor* of the state σ_2 if: (i) both σ_1 and σ_2 are in $\bar{\Sigma}(t)$; (ii) σ_1 can be reached from σ_2 in one single step: since χ_i is the last element of the σ_2 sequence, σ_1 is a neighbor of σ_2 if $\exists \chi_j \in X: \exists \lambda_{ij} \in A$ and $\sigma_1 = \langle \sigma_2, \chi_j \rangle$. The set of feasible neighbor of σ is called $N_{\sigma}(t)$.
- A *solution* Ψ is an element of $\widehat{\Sigma}(t)$ that meets some *exit condition* e .
- An *objective cost function* Γ_{Ψ} which assigns a cost value to each solution reached.

In other words the combinatorial optimization problem must be formalized in terms of shortest path problem through a graph $G = \langle X, A \rangle$, where each solution is represented by a sequence of components χ_i (path) satisfying the constrains $\Omega(X, A, t)$ and meeting some *exit condition* e . The ACO algorithms are implemented by building a colony of artificial ants (agents), that tries to approximate the optimal solution in the following way:

- Each agent has one *starting state* σ_s^k and one, or more, *exit condition* e^k .
- Each agent starts from its own starting state and moves through one of its own feasible neighbors, thus building a solution step by step. The processing ends when the agent reaches its own exit condition.

- Often a *heuristic function* η_{ji} is supplied, to indicate the convenience of moving from a state σ_j to $\sigma_i \in N_{\sigma_j}(t)$ based on a priori knowledge.
- Each connection λ_{ji} has an associated *pheromone trail* $\tau_{\lambda_{ji}}$ ¹ encoding the experience of the previous agents: the higher the pheromone deposit $\tau_{\lambda_{ji}}$, the lower the costs of previous agents' solutions that included connection λ_{ji} .
- Each agent on state σ_j moves towards one of its own feasible neighbors σ_i , through connection λ_{ji} , applying a *probabilistic rule*. The probabilistic rule is a function of: (i) the heuristic function η_{ji} ; (ii) the pheromone trail $\tau_{\lambda_{ji}}$; (iii) the information yield by the agent during its past travel to the current state.
- When leaving a state σ_j towards state σ_i via connection λ_{ji} , each agent may update the pheromone trail $\tau_{\lambda_{ji}}$.
- Once an agent has reached the exit condition it increases the quantity of pheromone on its path in a way inversely proportional to the cost of the solution it constructed. This is a *reinforcement learning* strategy (Sutton and Barto, 1998) that modifies the way the following agents will perceive the problem, thus performing a distributed adaptive approach.
- Since in this paper, the ACO algorithm is assumed convergent, when all agents has reached a solutions the pheromone trails are analyzed to find the best path that approximates the optimal solution of the combinatorial optimization problem. The convergence to the optimal solution of the ACO algorithms has been studied in several papers (Gutjahr, 2002; Stützle and Dorigo, 2002).

The next section shows how the path detection problem was mapped onto the ant colony optimization formalism.

3. Preprocessing

Before applying the ACO algorithm, we need to develop low-level preprocessing procedures to localize the optimal starting and final states and the appropriate local properties of road boundaries pixels.

3.1. Interest area: starting states, final states and set of components X

The final states and the set of components X are defined by the constrains of the output map. The vision sensor provides to the path planner a 50 m \times 50 m map of the space in front the vehicle, so 50 m represents the upper limit for the agents' movement. Each component $\chi_i \in X$ is a pixel between the upper limit and the bottom of the image. Consequently a feasible state σ is any ordered pixel sequence starting from the bottom of the image; a final state Ψ is any state ending on the upper limit.

¹ Or simply τ_{ji} .

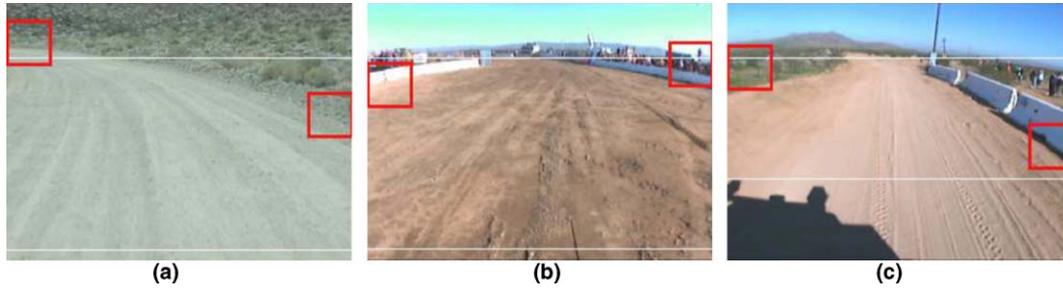


Fig. 3. Examples of different scenarios: the agents' starting areas are indicated by squares, the white lines indicate the area of interest.

The initial states are the pixels inside the two starting areas shown in Fig. 3 as squares. They are 40×40 pixels squares, placed in peripheries areas where a sufficient percentage of edges is present. In very sharp curves it is possible that only one boundary of the road is visible: in this case it is very helpful to move the starting areas: as shown in Fig. 3(a) the start and final states overlap, and only one bound will be detected. Therefore, before starting the execution, each agent of the colony is put randomly on a pixel in these areas.

As shown in Fig. 3(c) a vehicle shadow removal step is also implemented to reduce the processing area. Basically this procedure is based on the idea that the *vehicle* shadow is always placed on the bottom of the acquired images. Hence the image is analyzed to localize a region *connected to the bottom* of the image, whose brightness is significantly lower (in percentage) than the average brightness of the whole image. If such a region exists, it is used to define the lower limit of the area of interest. The shadow removal step is necessary since the shadow could generate strong edges and could lead to false values of the road chromatic properties.

3.2. Heuristic and cost function

An extremely important preprocessing step is the definition of the local properties that characterize the road boundaries in unknown and unstructured environments. To do this a special image representing distance to the average of the road, called *imD*, is built from the

320×240 acquired RGB image. The assumptions are: the vehicle is already on the road and the road is homogeneous.

The *imD* image is computed as follows: the 640×480 pixels image in the RGB color space is converted to a 320×240 pixels in a brightness independent domain: the RGB normalized (RGBN). Then a temporal average of red, green and blue values of road pixels is computed, and called $E_{road}(t)$: this depends on all the images acquired so far. The application of *vectorial euclidean distance function* $d(\vec{X}, \vec{Y})$ leads to a monochromatic image *imD*. Fig. 4 shows an example of RGBN image (a) and the corresponding distance to the average (b). Finally, applying a gradient operator, we obtain a monochromatic edge image (Fig. 4(c)), that represents the only a priori information given to the agents. The brighter a pixel, the higher the probability that it belongs to the boundaries of the road. This is due to the nice segmentation achieved in *imD* (Fig. 4(b)) that enhances the differences between pixels inside and outside the road, and, at the same time, smooths out the differences within those two classes. So, the attractiveness of a pixel is proportional to the brightness of its correspondent pixel in the edge image (this rule defines the *local heuristic function*) (Broggi et al., 2003). In the same way, the cost of the movement towards a pixel is inversely proportional to the brightness of its correspondent in the edge image (this rule defines the *cost function*).

Fig. 5 summarizes the whole block diagram of the preprocessing, where all the information needed by the ACO algorithm are depicted.

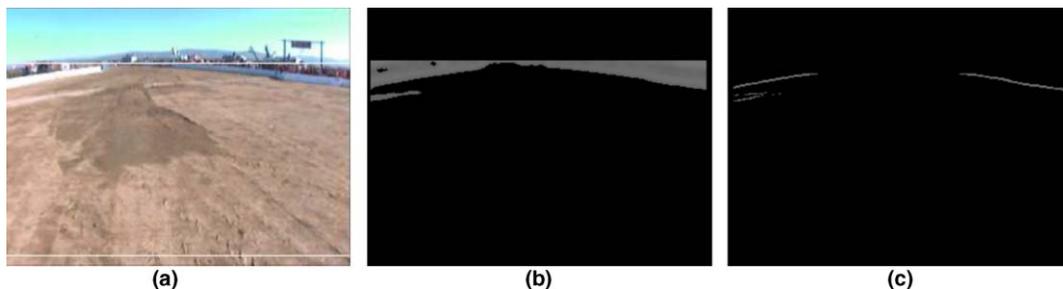


Fig. 4. RGB (a), distance to the average image (b), and a gradient operator computed to obtain the edge image (c).

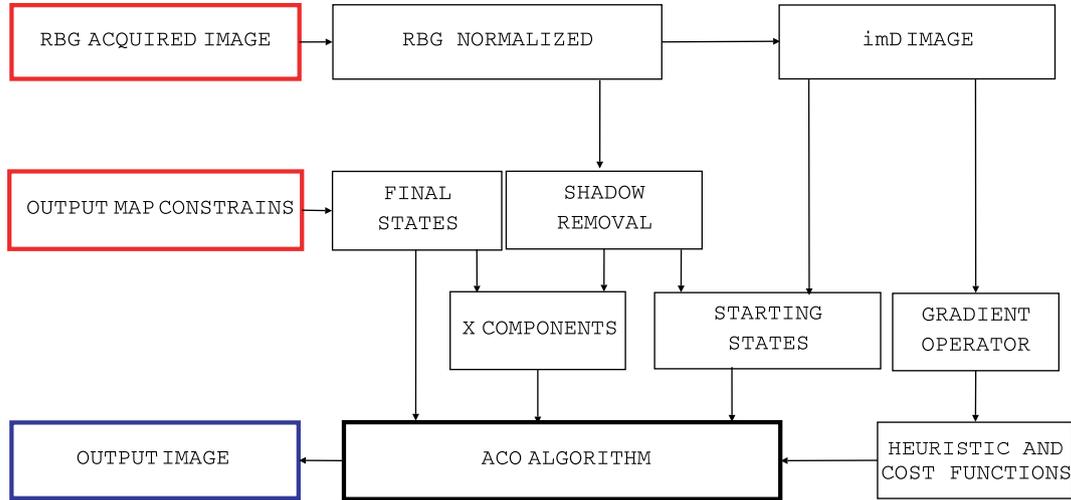


Fig. 5. Preprocessing block diagram.

4. Application of the ACO algorithm

4.1. The point of attraction

Before defining the moving rules, it is necessary to define the set of *feasible neighbors* N_{σ_h} of a state σ_h . Given a pixel (i_a, j_a) :

- An agent on $\sigma_h = \langle \sigma_{h-1}, (i_a, j_a) \rangle$ can move only towards the set of pixels of coordinates $\{(Center(P, i_a, j_a) - 3, j_a), \dots, (Center(P, i_a, j_a) + 3, j_a)\}$;
- in case of *backtracking* (see Section 4.2) the agents can move towards pixels $\{(Center(P, i_a, j_a) - 3, j_a - 1), \dots, (Center(P, i_a, j_a) + 3, j_a - 1)\}$;

where (i_a, j_a) are the row–column coordinates of the current pixel, and $Center(P, i_a, j_a)$ is a function of the current pixel and the *point of attraction* $P = (i_p, j_p)$. Assuming an orthogonal reference system in which the point of coordinates $(0, 0)$ is placed in the upper-left image corner, as shown in Fig. 6, the function $Center(P, i_a, j_a)$ is defined as follows:

$$Center(P, i_a, j_a) = \text{the nearest integer to } r(j_a - 1) \quad (1)$$

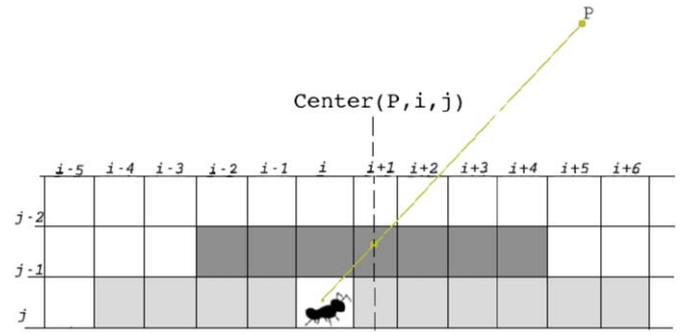
where $r(j)$ is the line connecting the point of attraction $P = (i_p, j_p)$ and the current position of agent (i_a, j_a) (see Fig. 6). The function $r(j)$ is computed as follows:

$$r(j) = i_p + \frac{i_a - i_p}{j_a - j_p} \cdot (j - j_p) \quad (2)$$

$j_a \neq j_p$ is always true by definition.

The point of attraction is computed frame by frame, using information on previously obtained road boundaries, as shown in Fig. 7.

The point of attraction polarizes the random moving component of the agents. Therefore, if the agents chose the next pixel in a *purely* random way, then the average moving direction of the colony is towards the point of

Fig. 6. The set of neighbors of (i, j) pixel: in light gray the neighbors in case of backtracking movement, in dark gray the neighbors in case of forward movement.

attraction (Fig. 8). The possibility of changing the point of attraction is of basic importance since the agents has limited capabilities of exploration, due to the limited set of feasible neighbors. Therefore it is necessary to provide some help to the agents in their searching movement, giving a preferential direction.

4.2. Motion rules

The motion rules can be divided into three levels. The first one is the well known *random-proportional* (Dorigo and Stützle, 2004) rule for artificial ants²:

$$a_{\chi_k} = \begin{cases} \frac{(\alpha) \cdot \tau_{hk}(t) + (1 - \alpha) \cdot \eta_{hk}}{\sum_{\chi_l \in \overline{N_{\sigma_h}}} (\alpha) \cdot \tau_{hl}(t) + (1 - \alpha) \cdot \eta_{hl}} & \text{if } \chi_k \in \overline{N_{\sigma_h}} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

² Hereinafter we will simplify the notation and indicate $\chi_k \in N_{\sigma_h}^*$ as $\chi_k \in X : \langle \sigma_h, \chi_k \rangle \in N_{\sigma_h}^*$, where $N_{\sigma_h}^*$ is any subset of N_{σ_h} .

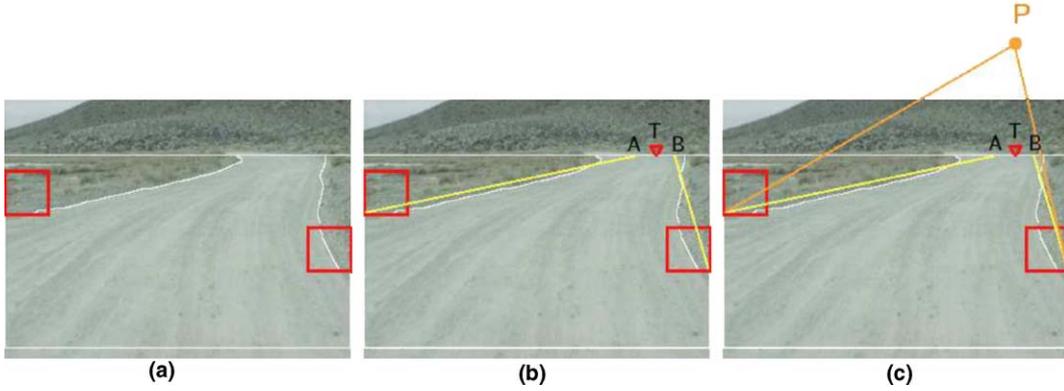


Fig. 7. Example of how the point of attraction is computed. The white curves on (a) represent the road boundaries obtained at time t , the straight lines in (b) represent their linear approximation and the triangle T is the midpoint between A and B . Thus the point of attraction at time $t + 1$ will have the same coordinate i_p of T at time t , while coordinate j_p is fixed for all possible points of attraction at any time.

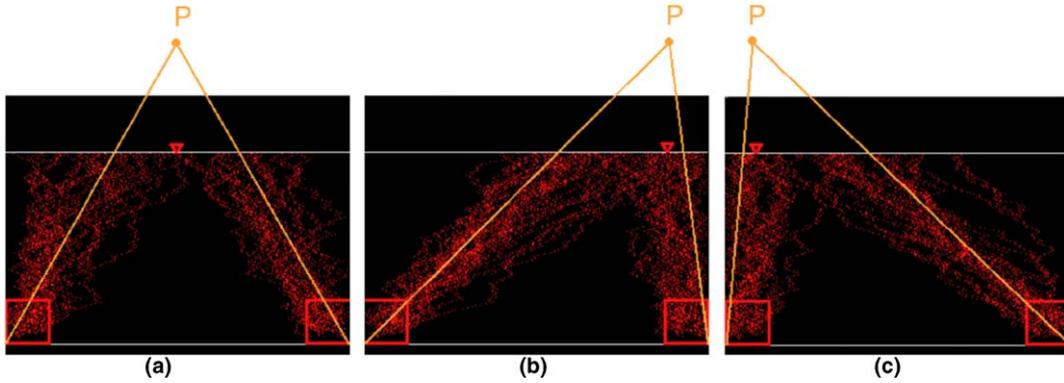


Fig. 8. Examples of the effect of different points of attraction P : the highlighted pixels represent the trails covered by the agents of each colony when moving randomly.

where:

- σ_h is the current state.
- a_{χ_k} is the probability of state $\langle \sigma_h, \chi_k \rangle$ of being the next state.
- $\overline{N_{\sigma_h}} = N_{\sigma_h} \cap \{(r, j - 1), r \in N\}$ are the neighbors on row $j - 1$.
- α is a parameter with which it is possible to tune the balance between edge-exploitation and pheromone-exploitation behavior of the agents.

The second level consists of the so-called *pseudo-random-proportional* rule, with which it is possible to improve the exploitation behavior of the agents:

$$\chi_{\text{next}} = \begin{cases} \chi_j : \eta_{\chi_k} = \max_{\chi_l \in \overline{N_{\sigma_h}}}(\eta_{\chi_l}) & \text{if } q \leq q_0 \\ \chi_k \text{ random based on Formula (3)} & \text{otherwise} \end{cases} \quad (4)$$

where q is a random variable with a uniform probability distribution, and q_0 is a threshold defined as follows:

$$q_0 = \gamma \cdot \frac{\max_{\chi_l \in \overline{N_{\sigma_h}}}(\eta_{\chi_l} - \eta_{\chi_{\text{next}}})}{\max_{\chi_l \in \overline{N_{\sigma_h}}}(\eta_{\chi_l})} \quad (5)$$

Parameter γ is used by the programmer to vary the balance between global *exploitation* and *exploration* behavior. If the

sum of pheromone trails and heuristic values of $\overline{N_{\sigma_h}}$ is zero, agents consider this movements too profitless, and perform backtracking steps. During backtracking agents do not proceed towards pixels on the upper row ($j - 1$), but towards pixels on the current row in $\widetilde{N_{\sigma_h}} = N_{\sigma_h} \cap \{(r, j), r \in N\}$. Each pixel $\chi_k \in \widetilde{N_{\sigma_h}}$ is chosen with a probability:

$$P_{\chi_k} = \begin{cases} \frac{\eta_{\chi_k}}{\sum_{\chi_l \in \widetilde{N_{\sigma_h}}} \eta_{\chi_l}} & \text{if } (\chi_k \in \widetilde{N_{\sigma_h}}) \wedge (\chi_k \neq \chi_h) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

This procedure could lead to a deadlock situation, so backtracking is permitted only once for each row. After this, a purely random move is generated. The main goal of this procedure is to speed up the convergence of the algorithm: even without backtracking, agents will be able to skip the regions with no pheromone and edges, but backtracking accelerates this process, allowing to decrease the number of agents needed to achieve the convergence.

4.3. Pheromone update

The N_a agents of a colony are divided into different subsets called $N_{a1}, N_{a2}, \dots, N_{an}$, each one characterized by dif-

ferent moving rules parameters, and executed in sequence. When every agent of a particular subset has reached the final pixels, the pheromone trails $\tau(t)$ are updated as follows:

$$\tau_{ji}(t+1) = (1 - \rho) \cdot \tau_{(ji)}(t) + \rho \cdot \sum_{k=1}^{N_{ax}} \Delta_{ij}^k(t) \quad (7)$$

where:

- t is a time measure, and represents the evolution of the pheromone deposit.
- $\rho \in (0, 1]$ is the pheromone *evaporation ratio*.
- Δ_{ij}^k is the k -ant contribution, computed as follows:

$$\Delta_{ij}^k = \begin{cases} Q/(L_k - L_{k\text{-best}}) & \text{if } k\text{-ant passed through } \lambda_{ij} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where Q is a fixed parameter. L_k is the cost of the k -agent, and $L_{k\text{-best}}$ is the cost of the current best solution respectively, computed as follow:

$$L_k = \frac{\sum_{ij \in \text{path}_k} (255 - \text{edge}_{ij})}{\text{path length}} \quad (9)$$

so the path cost is inversely proportional to the average brightness accumulated over the collection of its pixels: a path formed by bright pixels costs less than a path formed by dark pixels.

4.4. Management of agents

As anticipated, each colony has N_a agents, divided into several subsets $N_{a1}, N_{a2}, \dots, N_{an}$, each one characterized by different rules parameters:

- Agents in N_{a1} have $\alpha = 0$. From the analysis of Formulas (3) and (4) it is clear that this means that the agents movement is only based on heuristics, ignoring the pheromone deposit. This could be considered as an *edge-exploitation* phase, in which previous experience is not considered.
- The agents in subset N_{a2}, \dots, N_{an} have parameter α defined as follow:

$$\alpha_i = \frac{i}{\text{Number of subset}} \cdot \alpha_p \quad (10)$$

where α_p is a value defined by the programmer, constant for all executions. In this way, as the execution proceeds, agents become more and more sensitive to pheromone, and less to heuristics (*pheromone-exploitation*). The underlying ratio is that the pheromone deposit becomes more and more reliable as time goes by, because it represents the experience of a large number of agents.

Fig. 9 shows the agents' paths, represented by a sequence of highlighted pixels. Note that the paths are more concentrated on the actual road boundaries.

4.5. Solution extraction

To achieve the final solution two special agents, one for each colony, are created: they start from the starting areas and move pixel by pixel until a final pixel is reached. They are attracted *only* by the pheromone trails, and ignore the heuristic function.

The two solutions obtained in this way, one for each colony (left and right), will represent the final road boundaries. The two sequences of white dots shown in Fig. 10 represent the special agents' paths; the space between them

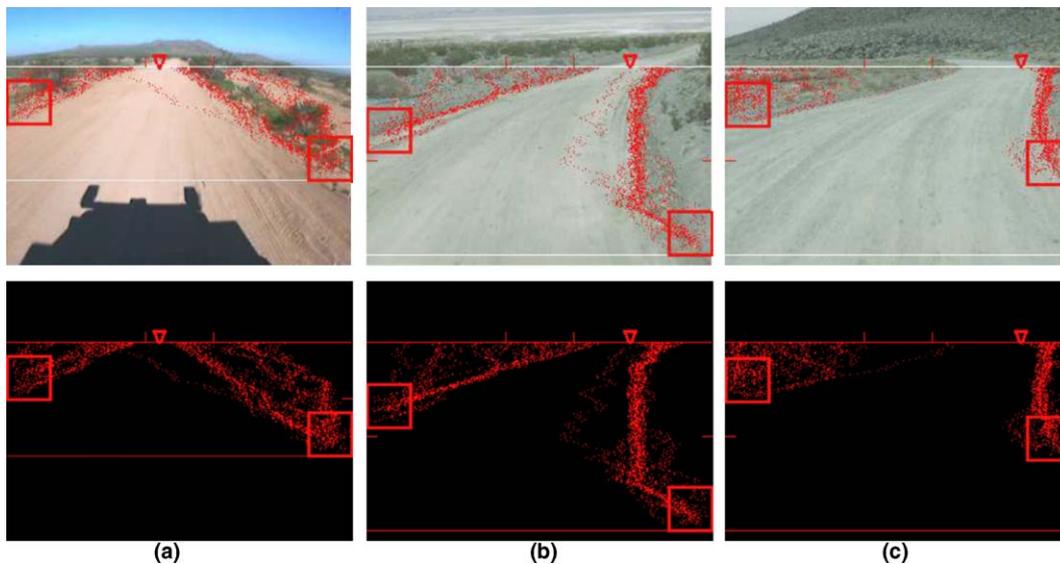


Fig. 9. The agents' paths: the upper row presents the original images with the agents' paths superimposed, the bottom row shows only the ants' paths for an easier visual evaluation.

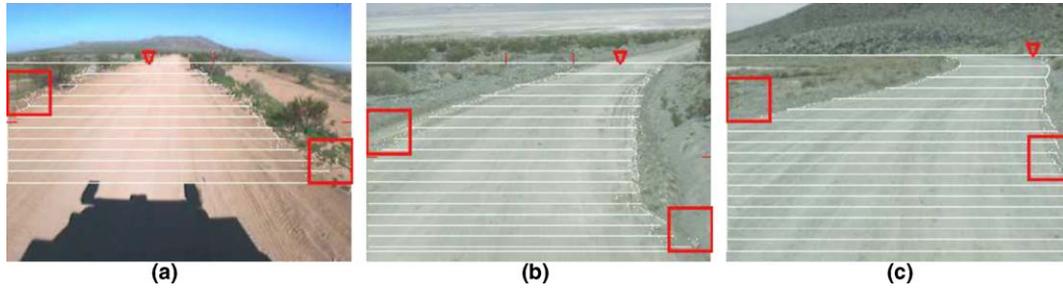


Fig. 10. Some results of correct path detection: the path is indicated by white horizontal lines. The vehicle shadow removal reduced the processing area in figure (a).

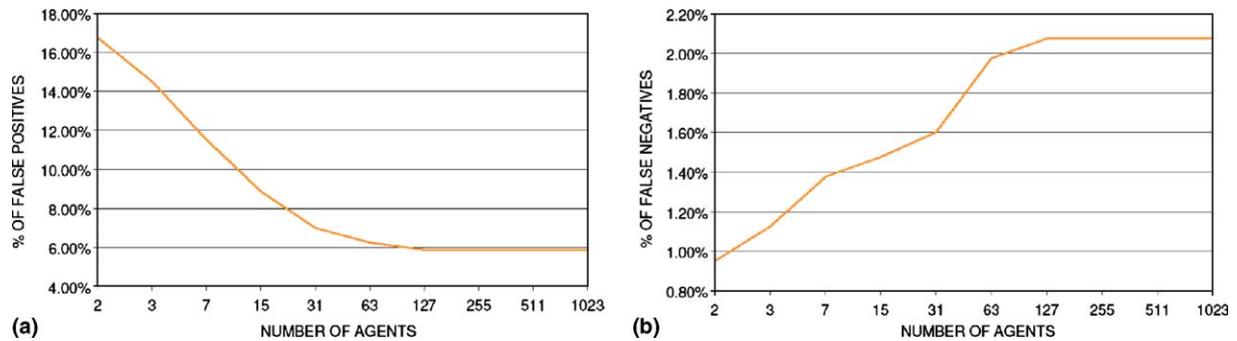


Fig. 11. Tests on the number of agents: false positives (a) and false negatives (b) percentage. The average values were computed for 100 runs of the algorithm. The subdivision of the subsets is made in the following way: given $N_a = 2^n - 1$, the subsets size has to be: $N_{1a} = N_a/2$, $N_{2a} = N_{1a}/2$, $N_{3a} = N_{2a}/2, \dots, N_{na} = 1$.

is considered as the surface of the road and represented by white lines.

5. Experimental results

There are many parameters that influence the algorithm behavior: the number of agents, the balance between exploitation and exploration, pheromone upgrade parameters, pheromone evaporation ratio, a few thresholds, etc. Some of these parameters has been chosen empirically to achieve the best results: pheromone evaporation ratio $\rho = 0.1$ in Formula (7); pheromone update contribution $Q = 1$ in Formula (8); in the shadow removal procedure, a region of pixels is considered as shadow when its brightness is $\leq 45\%$ of the average image brightness.

Some others parameters deserve a much more detailed discussion, since they are closely related to the multi-agent distributed nature of the proposed system. The first one, the number of agent for each colony, has also a great influence on the global performance. According to the order in which agents are executed (see in Section 4.4), we made some systematic tests, using presegmentated images.³ The human and algorithm's results were compared, measuring the error percentage. Fig. 11 shows the error percentage

as a function of the number of agents: *false positives* pixels are here defined as the ones classified as road by the algorithm but as off-road by the human operator, while *false negatives* are the pixels classified as off-road by the algorithm but as road by the human operator.

This test shows an interesting behavior of agents: when using a small number of agents the majority of the solutions is correctly attracted by areas with a high density of edges, but the results include many false positives and just a few false negatives. When raising the agents number, the agents tent to cover paths closer and closer to the road shape. Considering also the algorithm's time performance on the TerraMax hardware (shown in Fig. 12), the optimal number of agents is defined to be 63 for each colony.

Two more parameters that were tested are $\alpha_p \in [0, 1]$ and $\gamma \in [0, 1]$ defined in Formulas (3) and (4). The higher is α , the more agents are attracted by edges, and the lesser are attracted by pheromone; the higher is γ , the more agents have an edge-exploitation behavior, and the lesser have an exploratory behavior. Fig. 13 shows an interesting result: the error percentage is almost independent of α_p , while on the other hand it is highly dependent on γ . This is easily explained since α is ruled by the agents execution cycle (seen in Section 4.4): that mechanism prevents the disastrous effect obtained when all agents use a static $\alpha = 1$, that leads the agents to ignore the heuristic at all. The best results have been achieved with $\alpha_p \geq 0.8$. The great influence of γ shows the importance of the heuristic

³ There images were manually marked by a human operator that labeled road and off-road areas.

Number of agents	2	3	7	15	31	63	127	255	511	1023
Time (ms)	48	60	66	73	80	87	97	110	131	164

Fig. 12. Average execution time. Hardware: Intel Pentium IV 3 GHz, 1 GB RAM, Linux OS. Acquisition frame rate: 10 s^{-1} .

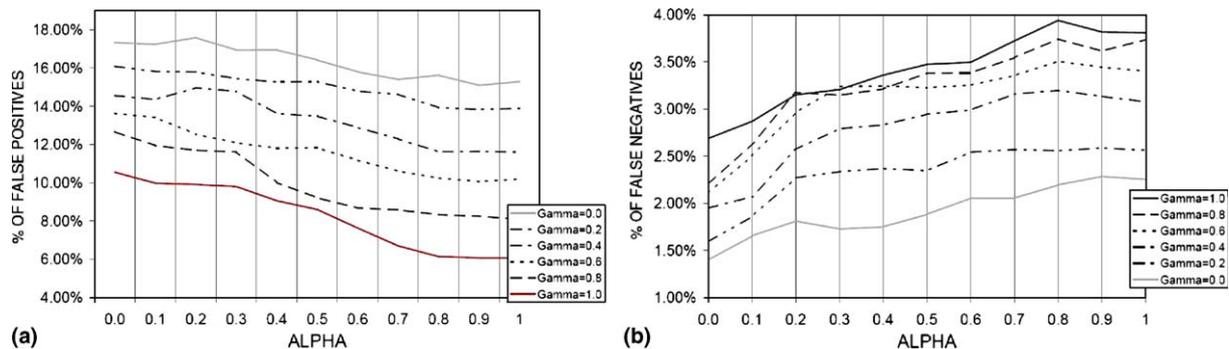


Fig. 13. Test on alpha and gamma tests: false positives (a) and false negatives (b) percentage. These test were performed under the same conditions described in Fig. 11.

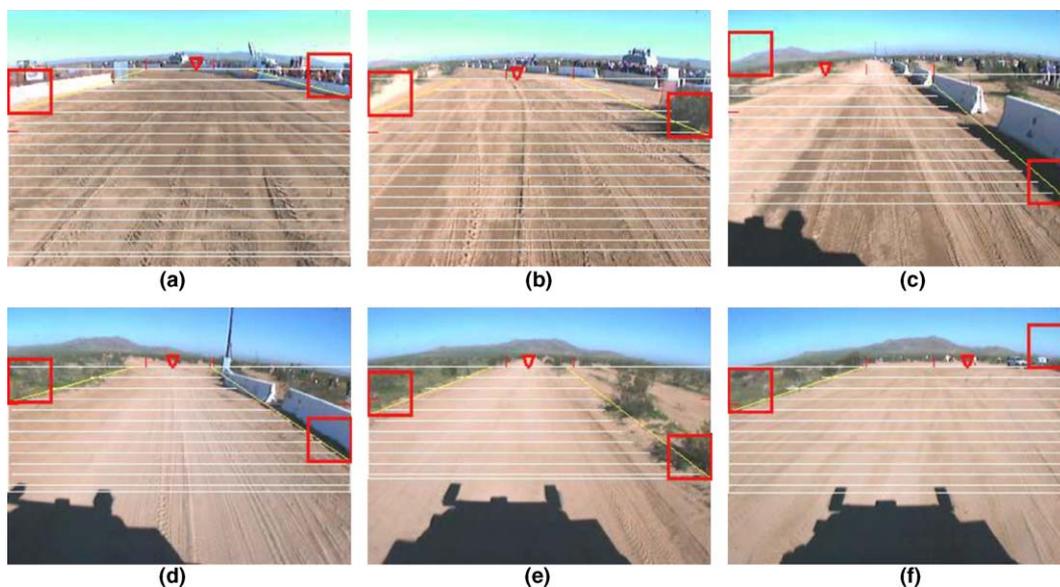


Fig. 14. Some results from the Grand Challenge 2004: the Grand Challenge starting area (a) and (b); the first curve (c); the first long straight road where some artificial boundaries are visible (c) and (d); a long straight road delimited by vegetation (e) and (f).

function, that represents the only knowledge of the agents about the problem. This can be considered as a proof of the highly significant preprocessing, since the rule in Formula (4) becomes more performing as it becomes more dependent on the heuristic function.

6. Conclusion

This paper presented an application of ant colony optimization to the road-borders detection problem, specifically developed for the DARPA Grand Challenge 2004

and tested on the Terramax autonomous vehicle. The ACO approach is well suited for a set of applications (for example the distributed network routing, or the Traveler Salesman Problem), but its application to solve the path detection problem was not straightforward. Therefore the basic ACO algorithm, well known from the literature, has been improved with some new features: polarization of the random movement by the point of attraction (Section 4.1), the mechanism which modifies the alpha parameter during the processing of each frame (Section 4.4), the ability of changing the starting states of agents frame

by frame (Section 3.1). These improvements complemented the ACO algorithm and made it suitable for the path detection problem.

The Artificial Vision and Intelligent Systems Lab of the University of Parma developed, for the **DARPA Grand Challenge 2004**, two more algorithms for path detection: one based on 3D models, one based on region growing using color and texture information. All these algorithms were tested on images taken during the DARPA Grand Challenge, and the ACO approach demonstrated to be robust in many critical situations. The main weakness which is still present in the proposed algorithm is the inability to detecting more than one pair of road-borders, which typically happens when approaching cross-roads or junctions. The next step, looking to the DARPA Grand Challenge 2005, will be allowing more than two agent colonies to detect all the road-borders candidates, and thus implementing a higher level processing to recognize what kind of path the vehicle is actually facing (single road, cross-roads, multiple lane road, etc.). Fig. 14 shows some real results.

References

- Bertozzi, M., Broggi, A., 1998. GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. Image Process.* 7 (1), 62–81.
- Bertozzi, M., Broggi, A., Fascioli, A., 2000. Vision-based Intelligent Vehicles: State of the art and perspectives. *J. Rob. Auton. Syst.* 32 (1), 1–16.
- Birattari, M., Caro, G.D., Dorigo, M., 2002. Toward the Formal Foundation of Ant Programming., in: *Ant Algorithms*, pp. 188–201.
- Broggi, A., Cellario, M., Lombardi, P., Porta, M., 2003. An evolutionary approach to visual sensing for vehicle navigation. *IEEE Trans. Ind. Electron.* 50, 18–29.
- DARPA, Grand challenge, available from: <<http://www.darpa.mil/grandchallenge>>.
- Dorigo, M., Stützle, T., 2004. *Ant Colony Optimization*. MIT Press, Cambridge, MA.
- Gómez, O., Barán, B., 2004. Reasons of ACO's Success in TSP., in: *Proc. of ANTS Workshop*, pp. 226–237.
- Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M., 1989. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* 76, 581–589.
- Gutjahr, W.J., 2002. ACO algorithms with guaranteed convergence to the optimal solution. *Inf. Process. Lett.* 82 (3), 145–153.
- Ozguner, U., Redmill, K.A., Broggi, A., 2004. Team TerraMax and the DARPA Grand Challenge: A General Overview, in: *Proc. of IEEE Intelligent Vehicles Symposium 2004*, Parma, Italy, pp. 232–237.
- Sim, K.M., Sun, W.H., 2003. Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Trans. Syst., Man, and Cybernetics, Part A* 33 (5), 560–572.
- Stützle, T., Dorigo, M., 2002. A short convergence proof for a class of ant colony optimization algorithms.. *IEEE Trans. Evol. Comput.* 6 (4), 358–365.
- Sutton, R.S., Barto, A.G., 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.
- Urmson, C., Anhalt, J., Clark, M., Galatali, T., Gonzalez, J.P., Gowdy, J., Gutierrez, A., Harbaugh, S., Johnson-Roberson, M., Kato, H., Koon, P.L., Peterson, K., Smith, B.K., Spiker, S., Tryzelaar, E., Whittaker, W.R.L., 2004. High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge 2004, Tech. Rep. CMU-RI-TR-04-37, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2004.
- Wade, A., Salhi, S., 2004. An ant system algorithm for the mixed vehicle routing problem with backhauls. *Metaheuristics: Computer Decision-Making*, 699–719.
- Yim, Y.U., Oh, S., 2003. Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* 4, 219–225.