An Evolutionary Approach to Visual Sensing for Vehicle Navigation

Alberto Broggi, *Associate Member, IEEE*, Massimo Cellario, Paolo Lombardi, and Marco Porta

***Abstract*—This paper presents an evolutionary approach able to process a digital image and detect tracks left by preceding vehicles on ice and snow in Antarctica. Biologically inspired by a colony of ants able to interact and cooperate to determine the shortest path to the food, this approach is based on autonomous agents moving along the image pixels and iteratively improving an initial coarse solution.**

The unfriendly Antarctic environment makes this image analysis problem extremely challenging, since light reflections, abruptly varying brightness conditions, and different terrain slopes must be considered as well.

The ant-based approach is compared to a more traditional Hough-based solution and the results are discussed.

***Index Terms*—Ant systems, automatic vehicle guidance, evolutionary algorithms, intelligent transportation systems (ITS), intelligent vehicles, machine vision.**

I. Introduction

THIS PAPER presents the artificial vision algorithms developed to autonomously drive a platoon of snowcats. This research is a part of the Ente per le Nuove tecnologie, l'Energia e l'Ambiente, Italy (ENEA), R.A.S. (Surface Antarctic Robot) Project, aimed at facilitating the transportation of people and goods in the Antarctica region during the Italian scientific missions at the South Pole. Researchers and equipment arrive from Italy via New Zealand on a ship which stops in the harbor of the "Baia Terra Nova" permanent Italian base, shown in Fig. 1.

Part of the people and goods must reach the "Dome Concordia" in-land Italian base, or the "Dumont d'Urville" French base, which are about 1200 and 2300 km far away from the harbor, respectively. This distance can be covered by helicopter, but goods generally travel by a platoon of snowcats, whose average speed is 10–12 km/h.

The R.A.S. Project was launched in order to partially automate the long and stressing procedure of people and equipment transportation in the unfriendly South Pole environment. In the first implementation the leading vehicle will be manually driven by an expert driver; in a second step, as shown in Fig. 2, it will be equipped with a camera and a TV antenna which will broadcast live images to the base, thus allowing remote driving.

Manuscript received September 3, 2001; revised May 10, 2002. Abstract published on the Internet November 20, 2002. This work was supported by the Ente per le Nuove tecnologie, l'Energia e l'Ambiente, Italy (ENEA) within the R.A.S. (Surface Antarctic Robot) Project.

A. Broggi is with the Dipartimento di Ingegneria dell'Informazione, Università di Parma, I-43100 Parma, Italy (e-mail: broggi@CE.UniPR.IT).

M. Cellario, P. Lombardi, and M. Porta are with the Dipartimento di Informatica e Sistemistica, Università di Pavia, I-27100 Pavia, Italy (e-mail: cellario@vision.unipv.it; lombardi@vision.unipv.it; porta@vision.unipv.it).

Digital Object Identifier 10.1109/TIE.2002.807688

All the remaining vehicles of the platoon will follow automatically in a train-like fashion, but, since cracks in the ice can put in serious danger both the driver and the snowcat itself, it is imperative that all vehicles follow the same precise path defined by the first vehicle. Since even small drifts from the original driving path defined by the human driver can be extremely dangerous, an extremely precise detection of the tracks left by the previous vehicle, a correct measurement of their position, and a smooth control of the actuators must be carefully designed, tested, and evaluated.

A preliminary test phase showed that the most promising sensor that should be able to deliver sufficiently precise measurements is a vision sensor (camera). Many other devices have been considered [1]–[3], even active ones, since the specific working site does not present any problem due to interference or to environmental pollution [4]. Anyway, vision seems the sensing capability that may deliver the highest performance in terms of precision of the localization.

Data are currently acquired from a monocular camera installed inside the driving cabin, but a stereo pair has been integrated inside the prototype vehicle in order to allow stereo image processing in the future. The calibration of the cameras is done thanks to a set of markers at known distances on a planar surface; Fig. 3 shows the test site in the Italian Alps and the grid used for calibration.

Besides cameras, a Global Positioning System (GPS) will be used for self-localization, but due to the impossibility of using Differential GPS (DGPS) in the area (no earth station is available and one cannot be maintained), its effectiveness will need to be confirmed. Fig. 4 shows the prototype of the vehicle follower developed within this project.

Due to the extreme conditions of the working environment, where temperatures can reach even $-80\ ^\circ$C, the terrain is completely covered by snow or ice, strong sun lighting and reflections may be present, and neither specific ground references are available nor assumptions can be made on the terrain slope, this application is extremely challenging and presents many additional problems with respect to the driving of unmanned vehicles on traditional structured roads [5]. For this reason an extremely careful analysis and design of the processing techniques is mandatory.

Several approaches have been considered due to the low visibility of white tracks on a white background, and specific filters have been developed in order to cope with the typical problems of this environment. The high problem complexity is slightly reduced by the low speed of the vehicle, which permits us to focus on the localization of tracks in a reduced close area only. Moreover, in the automatic driving of road vehicles [6] a special

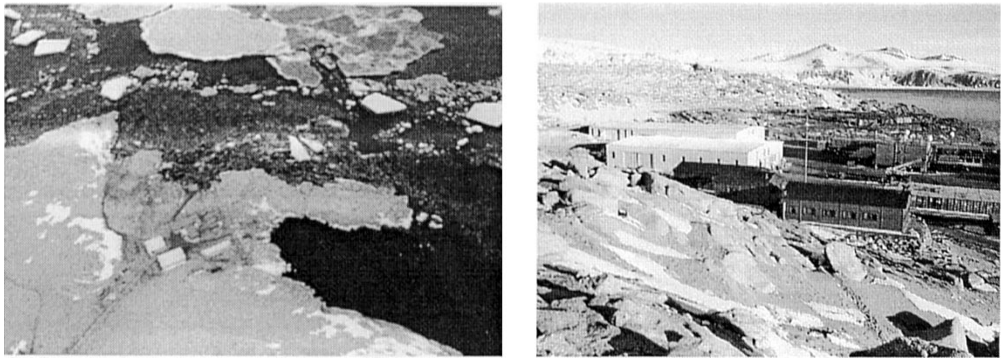


Fig. 1. Aerial views of the Italian base in Terra Nova, Antarctica.

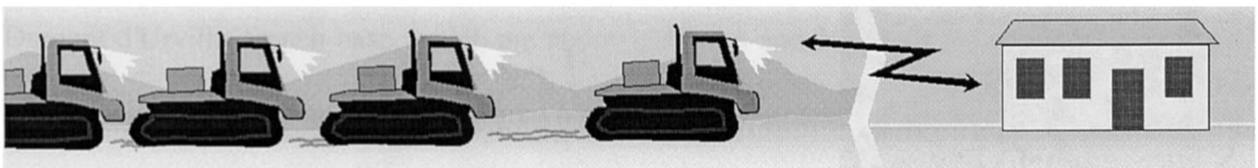
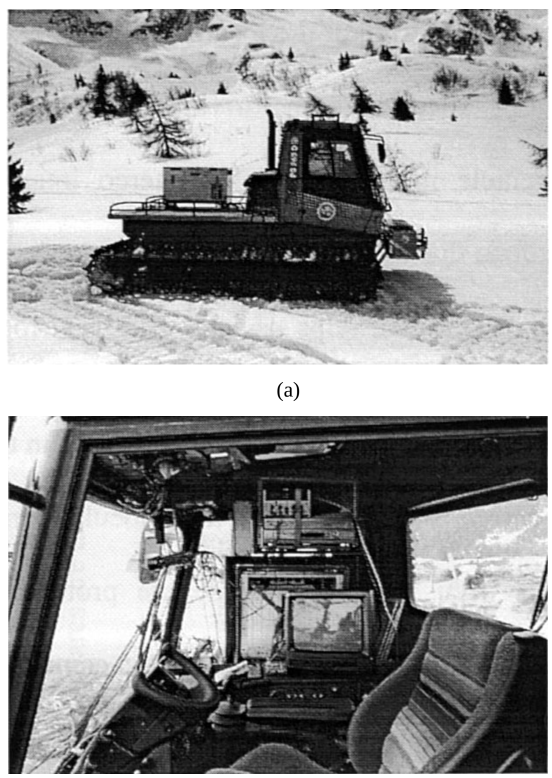


Fig. 2. Platoon of snowcat vehicles; the first will be either manually driven by an expert driver or remotely driven from the Italian base; all the others will follow automatically, using visual information only.

Fig. 3. Image of the test site in the Italian Alps; a planar patch of snow with markers is used to calibrate the cameras.



(a)

(b)

Fig. 4. Prototype of the automatic vehicle follower during a test at the Italian test site. (a) External view. (b) Internal equipment (stereo cameras are visible on the left side).

emphasis is generally given to the exploitation of *a priori* knowledge in order both to speed up the computation and to make the detection robust. In this case, only a little knowledge about the environmental conditions can be exploited: generally, no other vehicle or building is seen by the camera, and the only markings on the ice are due to the preceding vehicle. On the other hand, no assumptions can be made with respect to a possible flatness of the area ahead of the vehicle, nor to a given range of illumination of the scene. In other words, hilly conditions must be considered as well and, therefore, the camera orientation generally used in road environments (low toward the road ahead) cannot be replicated here. Besides the acquisition of a large amount of insignificant data during driving in flat areas, the framing of a large portion of the sky can raise another important problem: since in the working site the sun may be very low on the horizon, no specific camera orientation can overcome the problem of direct sunlight into the vision system. This is an extremely difficult issue that must be carefully considered in the development of vision algorithms.

The approach used to solve this artificial vision problem differs from traditional deterministic algorithms and was chosen to test the effectiveness of evolutionary approaches in this field. The proposed method falls into a class of nondeterministic algorithms based on natural metaphors that simulate the adaptive capabilities of natural systems. Essentially, this class of procedures tries to implement the natural engines of selection (corresponding to optimization) and mutation (corresponding to random search), through mechanisms of competition–cooperation within a population of agents (artificial ants, in this case).

Starting from the experience of the research group in the automotive field [7], a deterministic approach (based on the Hough transform) has been also developed and will be briefly described and compared with the evolutionary one. The deterministic algorithm derives from studies on lane detection developed within the ARGO project [8] by the same research group, aimed at the development of an intelligent vehicle able to drive autonomously in real traffic conditions and on real roads.

Unfortunately, the extremely different environmental conditions and precision requirements did not allow us to use the same algorithms verbatim, but the experience made on lane detection was of basic importance to the design of the low-level processing steps and the overall system integrated on the snowcat.

This paper is organized as follows. Section II will briefly present the evolutionary approach on which the tracks detection algorithm is based; Section III will describe the low-level portion of the image processing algorithm, while Section IV will present the details of the ant-based system. Section V will presents some results together with a critical analysis of the overall approach and a comparison with a more traditional Hough-based approach. Section VI will conclude the paper with final remarks and a description of current activities for the improvement of the complete system.

II. Evolutionary Approach

The Ant Colony Optimization (ACO) [9] is a distributed metaheuristic for hard combinatorial optimization problems, originally inspired by the communication behavior of real ants, based on pheromone (an odorous substance) as an indirect interactive medium. In nature, when ants find a new source of food, they communicate this information to other ants by marking their path with a pheromone trail, its intensity depending on the distance of the food source from the nest, and on its quality and quantity. Other ants are attracted by strong pheromone trails, thus, the path to an abundant food source close to the nest is marked again and again until it becomes more frequented and even more attractive.

This concept can be applied to hard combinatorial optimization problems by creating a colony of artificial ants that looks for optimal solutions in a constrained solution space. An artificial ant is essentially a simple agent that explores the solution space and builds partial solutions based on a local heuristic and on the information available from previous attempts of other ants (pheromone). The local heuristic guides the early stages of the computation, while the artificial pheromone trail accounts for the knowledge of the problem acquired since the beginning of the process, and becomes more and more important as a larger number of attempts is made.

The Ant System (AS) is the first algorithm belonging to this metaheuristic, and was applied to solve the classical Traveling Salesman Problem (TSP) [10], [11], an NP-hard combinatorial optimization problem. The TSP consists in finding the shortest path to visit a given set of cities. The AS starts by placing a number of artificial ants on randomly selected cities and then lets them run a complete tour. Each ant deposes a pheromone trail of intensity dependent on the length of its tour. The process is repeated for more cycles until convergence is reached. The original AS was further improved in the last few years, and gave rise to a family of algorithms that were experimented on solving problems of quadratic assignment [12], scheduling [13], vehicle routing [14], and network routing [15], among the others. See [16] for a recent survey.

The present work was inspired by an analogy between the TSP and the problem of vision-based track detection in digital images. A gray-level digital image is organized in pixels, where each pixel contains the information of light intensity in a gray scale code from 0 (black) to 255 (white). With a simple preprocessing step, the edges of objects can be extracted by calculating the two-dimensional spatial derivative of the intensity function. This step produces a new image, in which each pixel reports the value of the derivative normalized within the 0–255 range. Thus, values close to 255 indicate a strong derivative, i.e., a candidate for an edge, while values close to 0 indicate a homogeneous region.

In the case of tracks on the snow, no well-defined edge can be identified. The image resulting from the preprocessing step presents a sequence of disconnected regions of high derivative in correspondence of the disordered snow blocks moved by the snowcat.

The problem of recognizing the tracks is in fact reduced to that of tracing a consistent line from the lower border of the image up to the line of the horizon, that passes through as many edge-points as possible. In a sense, this problem is similar to the TSP, where the lower border is the starting city, the horizon line is the arrival city, and the edge-points are the cities to be visited. Once set in this framework, track detection can be accomplished by means of an algorithm based on the ACO approach, provided that appropriate heuristic and pheromone adjustment rules are designed. These considerations led to the design of a new algorithm, which will be identified from now on as AntPX (abbreviation of Ant-PiXel). While in the classical TSP the best solution is the shortest path, in AntPX artificial agents search the path that connects the highest possible number of edge points, under specific constraints.

Fig. 5 illustrates the natural metaphor underlying AntPX: the edge image under analysis functions as a two-dimensional map featuring the territory where the artificial ants move. Edge pixels represent ant-sized valleys and pixels belonging to homogeneous regions with no edges stand for the mountains.

The ants start from the bottom line (nest) and try to reach the top line (food) through the valleys. The ants of the first exploring group have no knowledge of the territory and choose

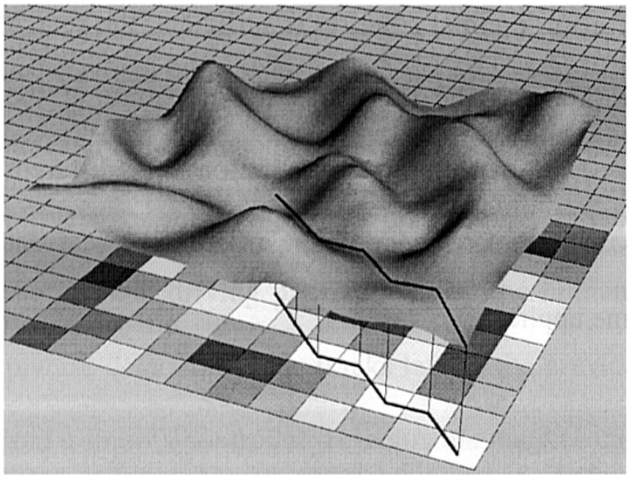


Fig. 5. Underlying natural metaphor of AntPX algorithm.

their path probabilistically, on the basis of the local heuristic of preferring valleys to mountains. When they reach the food line, a pheromone trail is deposed: the higher the number of edge pixels in the path, the stronger the pheromone trail left by a given ant. Following sets of ants observe the pheromone and trade their probabilistic decision with the experience of precedent ants. As more cycles are completed, ants pay attention to the accumulated knowledge (pheromone) besides the simple heuristic of following the valleys. Thus, the whole colony quickly finds a short way to reach the food line by passing through as many valleys as possible.

The ACO metaheuristic was chosen for a number of reasons. First of all, the edge image presents many noisy pixels due to the unpredictable nature of the environment. The stochastic choice of the next state in AntPX gives birth to a large number of alternative solutions. This feature, combined with the evolutionary feedback provided by the artificial pheromone, promises to be a robust way of avoiding false positive tracks. Secondly, there is no actual unique solution to the reconstruction of the track edges, but rather there is a region of good solutions that approximate the track edges with an acceptable tolerance. ACO algorithms have the desirable quality of converging quickly to a good subspace of the solution space. If a suboptimal solution is sufficient, the search process may be terminated after very few cycles. Finally, the intrinsic parallel nature of the algorithm may significantly improve the performance, by distributing computation over different parallel processors, or by parallelizing the processing thanks to parallel instructions of current processors; this feature can be significant for real-time applications.

III. Low-Level Preprocessing

Each gray-level image (256 $\times$ 200 pixels) undergoes a process of transformation before the analysis with the AntPX algorithm is carried out. The image needs a first low-level preprocessing step to prepare it for the following processing, and a second step to eliminate disturbing objects in the scope of the camera.

With respect to the first step, a 3 $\times$ 3 average filter is passed all over the image. After this step, a new image is created in which each pixel has an intensity value corresponding to the average of its own value and those of the eight pixels of the neighborhood. The immediate effect is to blur contours so that the edge extractor described below will produce wider edges, which proved to guide ants more effectively. At the same time, no clustering is applied in order to take full advantage of this effect. The second preprocessing step consists in the generation of a mask that restricts the area of analysis. In fact, the high-level track reconstruction phase, when run on the whole image, is generally disturbed by the presence of parts of the internal and external equipment of the snowcat (windscreen wiper, plow), and of possible shadows cast by mountains or by the vehicle itself, as shown in Fig. 6(a). These are dark items and have a strong contrast with the snow, thus producing strong edges that could distract the ant system algorithm. The problem is solved by drawing an intensity histogram of the image, that is a histogram of the number of pixels showing a given gray level [see Fig. 6(b)]. Pixels having an intensity value below a given threshold do not belong to snowy regions and are masked out from the interesting area.[1] The result of this operation is shown in Fig. 6(c).

Two further steps are taken before applying the AntPX algorithm. First, the two-dimensional spatial derivative of the intensity image is calculated. Two 3 $\times$ 3 isotropic filters are applied to the image, in order to extract the horizontal and vertical component of the gradient. The isotropic operator was chosen to provide accurate information on the phase of the gradient, which is useful to guide the artificial ants during the recognition step as it will be clear below. The result of this step is an image featuring the edges of the original one encoded in a 256 gray scale [Fig. 6(d)]. Finally, the edge image is normalized to cover the full range of the 0–255 gray scale and subsequently equalized. These operations improve the performance of the AntPX algorithm by enlarging and discretizing the spectrum of the gradient values that guide the ants' local heuristic search.

IV. High-Level Processing

The AntPX algorithm is here introduced within the ACO metaheuristic framework, through the AS algorithmic instance; a consistent theoretical and experimental reference is thus provided for the analysis of AntPX structure, of its relation with existing literature, and of its possible evolution directions.

A. ACO Metaheuristic

As previously introduced, ACO is a distributed metaheuristic approach for solving hard combinatorial optimization problems, based on the indirect communication of a colony of simple agents and inspired by real ants' communication behavior; pheromone trails in fact provide distributed numerical information used by ants to probabilistically construct solutions and adapted during the algorithm's execution to reflect the ants' experience.

Artificial ants in ACO implement a stochastic construction heuristic by making probabilistic decisions on the basis of artificial pheromone trails and possibly available problem-specific heuristic information. The stochastic component in ACO allows

[1]Masking based on a fixed threshold has been demonstrated to be reliable on many different sequences of interest.

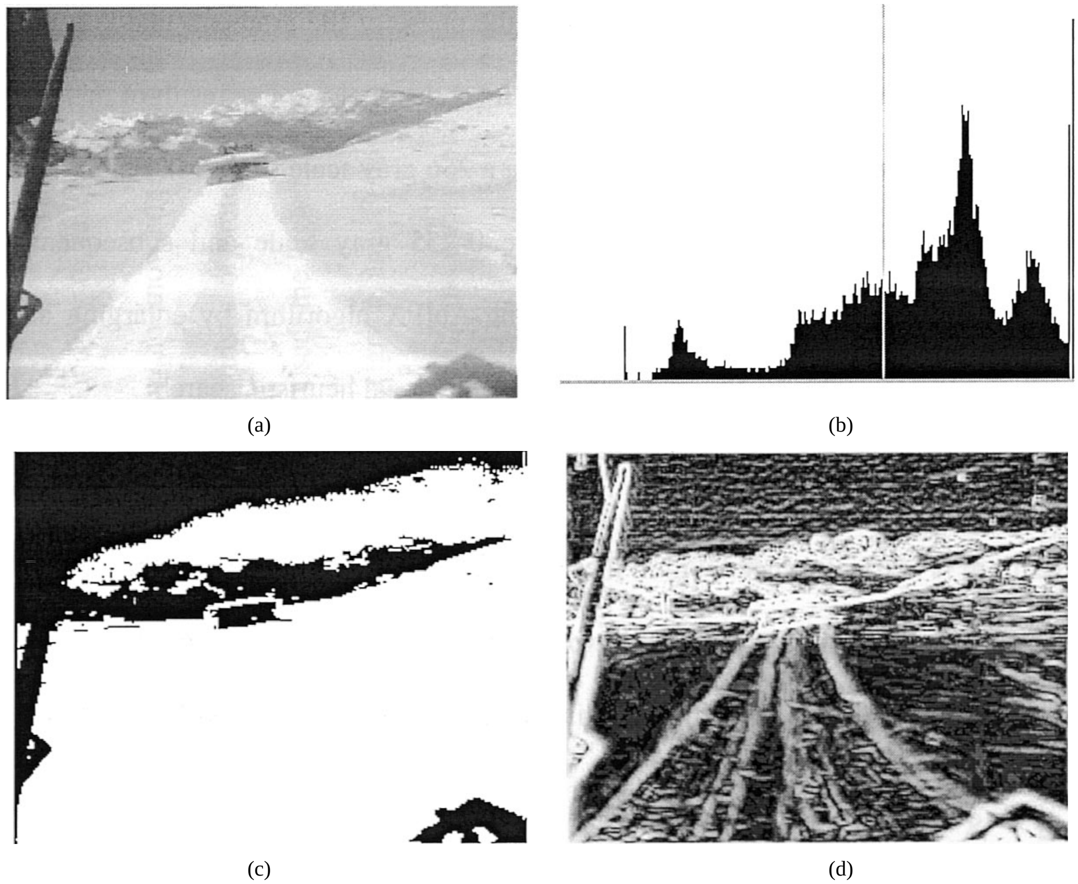


Fig. 6. (a) The low-level preprocessing steps starts from a typical image of the test sequence, containing a shadow and the windscreen wiper; (b) a gray-level histogram is computed for each image, the vertical line indicating (c) the threshold for the definition of the mask; (d) edges are extracted with an isotropic filter—brighter pixels indicate a higher derivative. The AntPX algorithm will analyze the edge map (d) only in the bright regions of the mask (c).

the ants to explore a larger number of solutions than greedy heuristics; at the same time, the use of heuristic information can guide the ants toward the most promising solutions. Additionally, the collective interaction of a population of agents (colony) can result in an increased algorithm robustness and efficiency.

In principle, ACO can be applied to any discrete optimization problem for which some solution construction mechanism can be conceived. A generic problem representation of a combinatorial optimization problem which can be exploited by the artificial ants includes the following:

- a set X of *components* χ_i;
- a set Λ of *connections/transitions* λ_{ij} among *components* χ_i and χ_j;
- a set Σ of possible *states*, defined as variable-length sequences of *components* $\sigma = \langle \chi_i, \chi_j, \ldots \rangle$;
- a set Ω of *constraints*;
- a set Σ^* of *feasible states*, defined by constraints in Ω, with $\Sigma^* \subseteq \Sigma$;
- an *objective cost function* φ which assigns a cost value to each state $\sigma \in \Sigma$.

Components $\chi_i \in X$ and connections $\lambda_{ij} \in \Lambda$ may have an associated *pheromone trail* τ (τ_i if associated to components, τ_{ij} if associated to connections) encoding a long-term memory about the ants' search experience, and a *heuristic value* η (η_i and η_{ij}, respectively) representing *a priori* information about the problem instance (e.g., an estimate of the cost of extending the current state).

The goal is to find a *globally optimal* solution σ^*_{opt}, that is, a minimum cost state that satisfies constraints in Ω. Given the above problem representation, artificial ants build solutions by moving through adjacent states of the problem on an associated construction graph $G = (X, \Lambda)$, where nodes are components in X and arcs are connections in Λ.

More precisely, each ant k of the colony is given a memory used to store information about the followed path[2] and is assigned a *start state* σ_s^k and one (or more) *termination conditions*; ant k thus selects the next move by applying a probabilistic decision rule, function of problem constraints and locally available pheromone trails and heuristic values.

Once an ant has built a solution, it evaluates the found sequence and deposits pheromone on the components or connections it used; this pheromone-coded information will direct the search of upcoming ants, in a sort of distributed reinforcement learning process [17] in which the single agents are not adaptive themselves but adaptively modify the way the problem is represented and perceived by other agents.

Besides *ants' activity*, an ACO algorithm includes two additional procedures: *pheromone trail evaporation* and (optional) *daemon actions*.

[2]Memory can be used to implement constraints, to evaluate the solution found, and to eventually retrace the path backward to deposit pheromone.

Pheromone evaporation (i.e., trail intensity decrease over time) is needed to avoid an unlimited accumulation of pheromone and the possible resulting premature convergence toward a suboptimal solution region; it thus implements a useful form of *forgetting*, which favors the exploration of the search space.

Daemon actions may be used to perform centralized actions which cannot be implemented by single agents, as the collection of global information used to bias the search process from a nonlocal perspective. The metaheuristic does not specify how these three procedures (*ants' activity*, *pheromone evaporation*, and *daemon actions*) should be scheduled or synchronized: designers are therefore free to specify the way in which these procedures should interact.

B. AS Algorithmic Instance

The Traveling Salesman Problem, a paradigmatic NP-hard optimization problem, has often been used as a common reference to test AS-related ideas and algorithmic variants [18].

In AS each ant is initially put on a randomly chosen city and iteratively moves from city to city. In particular, an ant k in city i chooses to go to a still unvisited city j with a probability $p_{ij}^k(t)$, defined as

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum\limits_{l\in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, & \text{if } j \in N_i^k \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where N_i^k is the feasible neighborhood of ant k (that is, the set of cities to be visited), $\eta_{ij} = 1/d_{ij}$ is available heuristic information (d_{ij} being the *a priori* defined distance between city i and city j), α and β are parameters which determine the relative influence of pheromone trail and heuristic information; parameters α and β have a strong influence on the algorithm behavior: with $\alpha = 0$ only local heuristic is considered, implementing a classical stochastic greedy; with $\beta = 0$ only pheromone information is considered, leading to the rapid emergence of a *stagnation* situation, generally within a suboptimal solution.

After each ant has completed a tour (that is, it has constructed a possible solution), pheromone trails are updated by first letting the pheromone evaporate and then allowing each ant to deposit its pheromone contribution on the arcs belonging to its path

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k(t) \tag{2}$$

where m is the number of ants and $\rho (0 < \rho \leq 1)$ is the pheromone trail evaporation rate. $\Delta\tau_{ij}^k(t)$ is the amount of pheromone ant k deposits on arcs λ_{ij}, defined as

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L^k, & \text{if arc } \lambda_{ij} \text{ is used by ant } k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where Q is a predefined constant and L^k is the path length of ant k (defined as the sum of each visited arc length d_{ij}). Arcs used by many ants and contained in shorter paths will receive more pheromone and will therefore be more likely to be chosen again in subsequent iterations. Algorithm convergence is thus reached through an *autocatalytic* process, which exploits search experience *positive feedback*.

For best performance, an ACO-based algorithm has to achieve an effective balance between the *exploitation* of the search experience and the *exploration* of the search space; such a balance may be achieved through an accurate pheromone trail management.

On the one hand, search space *exploration* (primarily achieved by the ants' stochastic solution construction) may be encouraged by the introduction of upper and lower bounds ($\tau_{\min}$ and $\tau_{\max}$) for pheromone trail values τ_{ij} and by explicit run-time reinitializations of their values (to $\tau_{\max}$ for instance) [12], thus trying to avoid stagnation situations and premature convergence, thus maintaining a minimal level of exploration of the search space.

On the other hand, search experience *exploitation* is simply favored by making the pheromone update a function of the solution quality achieved by every single ant, as indicated in (3). Exploitation is additionally favored over exploration by the introduction of an *elitist strategy* [19], an instance of *daemon action*, consisting in an extra deposit of pheromone $\Delta\tau_{ij}^{k-best} = Q'/L^{k-best}$ over the points of the best path (performed by ant *k-best*) since the start of the algorithm (*global-best*) or within the current iteration (*iteration-best*); a stronger elitist strategy may allow only the best ant to update pheromone trails.

An effective exploitation of heuristic and pheromone-coded knowledge can also be achieved, during solution construction, by applying the so-called *pseudorandom proportional* rule [18], that is, an ant k in city i moves to city j, using the following action choice rule:

$$j = \begin{cases} \arg\max\limits_{l\in N_i^k} \left\{ [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta \right\}, & \text{if } q \leq q_0 \\ P, & \text{otherwise} \end{cases} \tag{4}$$

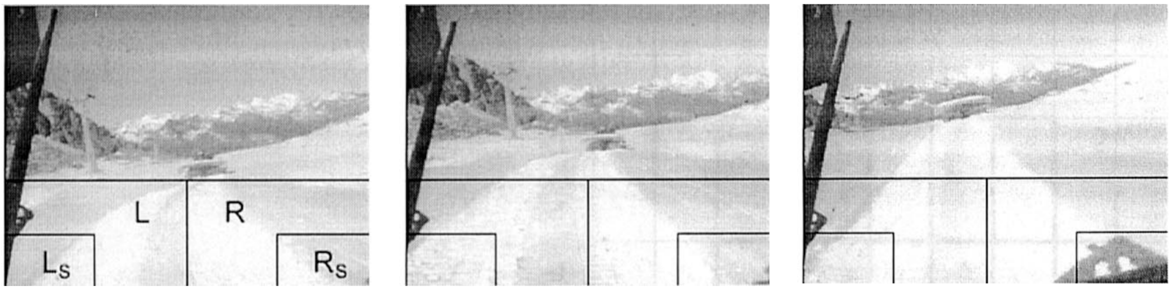
where q is a value chosen randomly with uniform distribution in [0,1], q_0 $(0 \leq q_0 \leq 1)$ is a predefined parameter, and P is a random variable selected according to the probability distribution $p_{ij}^k(t)$ defined in (1); hence, with probability q_0 ant k moves to the city j for which the product between pheromone trail and heuristic information is maximum, while with probability $(1 - q_0)$ it performs a biased exploration, based on the probability distribution $p_{ij}^k(t)$. Exploitation is thus favored over exploration, with parameter q_0 values close to 1, while $q_0 = 0$ evidently implements basic AS decision rule (a *random-proportional* rule).

Finely tuned variations of parameters α and β may also be effectively used to relatively balance the importance of exploitation of τ and η individually, within the overall exploitation strategy.

Other interesting improvements to basic AS algorithms may be introduced by coupling them with specific local search algorithms, which locally optimize ants' solutions subsequently used in the pheromone update (another instance of daemon action).

C. AntPX Algorithm

Current research is increasing the number of problems successfully solved by ACO algorithms, including real-word industrial applications. Though rather successful on well-defined combinatorial optimization problems, ACO algorithms



(a) (b) (c)

Fig. 7. Left and right starting points areas (L_S, R_S), within respective left and right rectangles of analysis (L, R).

seem to provide significant potential when applied to ill-structured problems for which effective local search techniques are not available. Starting from AS, new algorithms have been proposed that, though less biologically inspired, still preserve many basic features of the original approach: a cooperative colony of agents, a probabilistic solution construction procedure based on artificial pheromone trails and local heuristic information, pheromone evaporation and reinforcement (guided by solution quality), positive feedback, and autocatalysis.

The AntPX Algorithm has been developed within this research direction, in the attempt to solve ill-conditioned problems in the computer vision domain; details about the AntPX Algorithm are now introduced on the basis of the ACO metaheuristic, with particular reference to the AS algorithmic instance, as introduced above.

An ACO-adapted problem representation of the track detection problem includes a set X of components χ_i represented by pixels in the image under analysis, a set Λ of connections λ_{ij} among pixels, a set Σ of possible states σ represented by ordered sequences of pixels, and a set Σ^* of feasible states σ^*, as defined by a set Ω of problem constraints, directly implemented in the artificial ants' searching behavior, as described later.

The overall goal is to find a globally optimum solution σ^*_{opt}, defined as a consistent path from the lower to the upper border of the image, indicating the most probable location for the snow track being detected.

Heuristic values $\eta_i = 1/(d_i + 1)$ are associated to each pixel component χ_i, representing locally available information, where $d_i = 255 - edge(x, y)$, and $edge(x, y)$ is the gray-level value of pixel (x, y) in the edge image being analyzed.[3]

Pheromone trails values τ_{ij} are associated to each connection λ_{ij} among pixel components, encoding the long-term ants' knowledge, updated by the ants themselves (and by daemon actions).

This problem representation is exploited by the artificial ants to build solutions, by moving through feasible sequences of pixel on an associated graph ($G = X, \Lambda$), where nodes are pixels and arcs are transitions among pixels.

More precisely, each ant is given a memory to store private information about the followed path, which can be used later to evaluate the solution found and to retrace the path for pheromone laying; each ant k is also given a start state σ^k_s and a termination condition, that is, a start and an end set of pixels, within a predefined area of analysis, as described below.

As indicated in Fig. 7(a), each track has an associated independent rectangle of analysis whose size has been fixed (128 $\times$ 80 pixels) according to problem context knowledge: though terrain-induced vibrations and oscillations may in fact introduce significant noise to the actual area occupied by tracks within the whole image [Fig. 7(a)–(c)], the predefined area of analysis will always include a significant portion of tracks.[4]

Fig. 7(b) and (c) also reveals that the lower portions of tracks do not always start from the lower border of the image, but sometimes even from the lateral side, though still remaining confined in the two lower corners.

An ant k is therefore randomly assigned a starting point within the starting corner relative to the track being searched for; in case of shadows extending in the corner areas [Fig. 7(c)], the masked-out shadowy regions must be excluded from the set of feasible starting points: this can be easily accomplished by systematically moving the candidate starting point until it has reached an unmasked pixel or, eventually, the predefined bounding perimeter of the starting area (64 $\times$ 40 pixels). Each ant is thus initially put on a randomly chosen pixel in the start area and then iteratively moved from pixel to pixel, row by row, toward the upper border of the search rectangle; this row-by-row movement constraint eliminates the need to check for next-state feasible row coordinates; additionally, as described in Fig. 8, each ant is allowed to move horizontally within a limited feasible neighborhood of seven columns only, centered around a next-state reference pixel, whose coordinate shift is calculated on the basis of edge orientation in the original image (that is, its gradient phase value, precalculated during low-level processing).[5]

Each ant then selects the move by applying a probabilistic decision rule, on the basis of locally available pheromone and heuristic values, τ_{ij} and η_j, respectively; in particular, an ant k in pixel i decides to go to pixel j with a probability $p^k_{ij}(t)$, defined as

$$p^k_{ij}(t) = \begin{cases} \dfrac{\alpha \cdot \tau_{ij}(t) + (1-\alpha) \cdot \eta_j}{\sum\limits_{l \in N^k_i} [\alpha \cdot \tau_{il}(t) + (1-\alpha) \cdot \eta_l]}, & \text{if } j \in N^k_i \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

[3]The [0,255] intensity range of the preprocessed edge image is remapped in d_i as a [255,0] range of gray-level distances from the absolute white (255); in η_i calculation, d_i are incremented by 1 to avoid division by zero.

[4]An adaptive search area (e.g., based on preceding vehicle detection or on vanishing point calculation and prediction) will overcome these environment-related problems in future improvements of the algorithm.

[5]The reference point column shift is discretized and limited to a range of seven pixels centered on the current column.

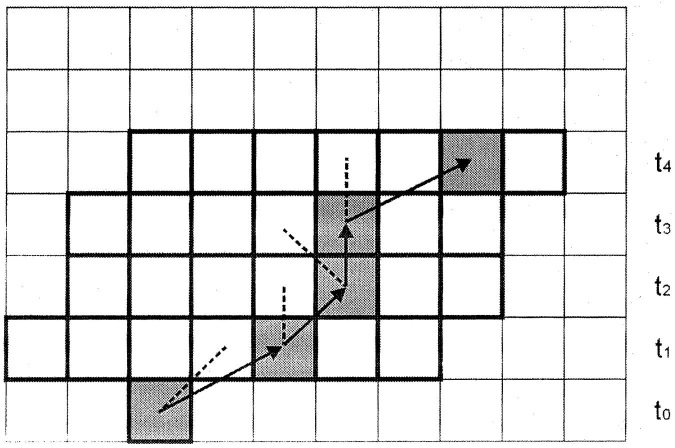


Fig. 8. An ant's row-by-row movements (gray squares), within the image matrix; each ant is allowed to move horizontally within a limited neighborhood of seven columns (bold squares), centered around a next-state reference pixel, whose coordinate shift is calculated on the basis of edge orientation in current pixel (dotted line).

where N_i^k is the feasible neighborhood of ant k described above, and α is a parameter which determines the relative influence of pheromone trail and heuristic information. The above formula differs from the classical one given in (1), by the absence of a second independent parameter β and for an overall computational simplification, particularly useful for real-time applications.

Once every ant has completed its tour (that is, it has constructed a feasible solution), pheromone trails are updated, through evaporation and reinforcement, in analogy with (2) in the basic AS algorithm

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\sum_{k=1}^{m}\Delta\tau_{ij}^k(t). \quad (6)$$

This expression introduces an additional influence of parameter ρ with respect to $\Delta\tau_{ij}^k(t)$ ants' contributions, which, in turn, are defined as

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/(L^k - L^{k-best}+1), & \text{if arc } \lambda_{ij} \text{ is used by ant } k \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where Q is a predefined constant and L^k is the path "length" of ant k, re-defined as the sum of each visited node's gray-level value d_i, divided by the actual number of visited pixels (that is, the average gray-level value of the path effectively followed);[6] L^{k-best} is the best L^k value found within the current iteration (*iteration-best*).[7] In general, arcs used by many ants and contained in short paths will receive more pheromone and will, therefore, be more likely to be chosen again in future iterations. As for the basic AS algorithm, convergence is thus reached through an autocatalytic process, which exploits search experience positive feedback.

As previously discussed, exploitation of the ants' search experience is favored over space exploration by making the pheromone update a function of the solution quality, as in (3); exploitation is additionally favored in (7) by the introduction of an implicit *elitist strategy*, consisting of a pheromone deposit balanced on the relative difference $(L^k - L^{k-best})$. Between the current and the best solution found: an instance of a *daemon action*, used to bias the search from a nonlocal perspective.

An additional exploitation of locally available knowledge has been also implemented by applying a variation of the so-called pseudorandom-proportional rule, introduced in (4): an ant k in pixel i first decides to move to pixel j with probability $p_{ij}^k(t)$ indicated in (5); then, *a posteriori*, it decides whether to accept the candidate pixel j or to eventually select another neighbor pixel that maximizes the heuristic knowledge η_j, by applying the following redefined action choice rule:

$$j = \begin{cases} \arg\max_{l \in N_i^k} \eta_l, & \text{if } q \le q_0 \\ P, & \text{otherwise} \end{cases} \quad (8)$$

where q is a value chosen randomly with uniform distribution in [0,1], P is a random variable selected according to the probability distribution $p_k^{ij}(t)$ defined in (5), and q_0 ($0 \le q_0 \le 1$) is an adaptive parameter, defined by

$$q_0 = \gamma \cdot \frac{\max_{l \in N_i^k} \eta_l - \eta_j}{\max_{l \in N_i^k} \eta_l} \quad (9)$$

$\gamma (0 \le \gamma \le 1)$ being a predefined parameter and η_j the heuristic value associated with the next-state candidate pixel j, as proposed by (5).

As previously stated, there is no actual true solution to the tracks detection problem; rather, there is a region of good solutions that approximates the track edges with acceptable tolerance; algorithm termination is thus reached when the majority of ants concentrate in a region along the same path or when a maximum number of iterations has been reached.

Eventually, the best solution is found by analyzing the pheromone matrix and extracting the maximum pheromone values from each row, as depicted in Fig. 9: the overall path will thus be composed of the most frequented pixels[8] in the last few cycles, which catalyzed the best ants' tours.

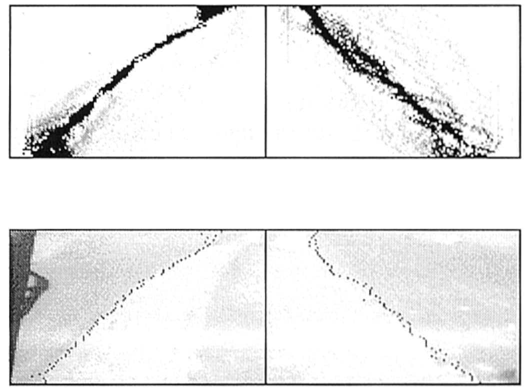
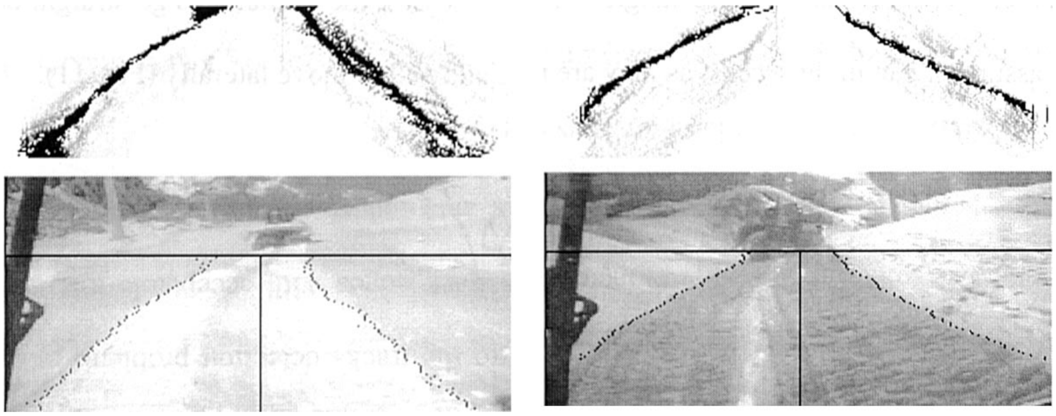


Fig. 9. Pheromone matrix visualization (above) and relative solution extraction (below); the maximum pheromone values are extracted from each row and their corresponding pixels are marked in black onto the original image.

[6]Each ant is, in fact, allowed to construct solution paths of different absolute length, depending on the particular starting point: normalization against the actual number of visited pixels is therefore required for a correct comparison among the ants' paths.

[7]$(L^k - L^{k-best})$ difference is incremented by 1 in (7) to avoid division by zero.

[8]Instead of the "shortest" visited path, as in the original AS algorithm previously introduced.



(a) (b)

Fig. 10. The AntPX algorithm is designed to follow tracks that contrast with the surrounding snow, both in the case of (a) bright tracks on darker snow and (b) on the opposite.

Optimal parameter values for algorithm convergence are generally sensitive to the specific problem–algorithm combination and most of the times they must be set experimentally.

Though ACO algorithms seem to be rather robust to the actual number of ants used, $m = 64$ has been used in this instance of AntPX, that is, a value close to the average number of pixel-components to be visited by ants.[9]

Tested values for parameters included the following: $Q = \{1, 255\}$, $\alpha = \{0, 0.5, 1\}$, $\rho = \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and $\gamma = \{0, 0.5, 1\}$; preliminary investigations encouraged toward the following parameters setting: $Q = 1, \alpha = 0.5, \rho = 0.1, \gamma = 1$, whose associated results are presented in the next section.

V. Experimental Results and Critical Analysis

Experimental images show many interesting situations, with or without noise. Fig. 10 illustrates two typical frames, in which, according to the position of the sun, tracks are characterized as bright regions on a dark background (sun behind the vehicle) or as dark regions on a bright background (sun in front of the vehicle). To our purposes, however, only the relative contrast between the tracks and the rest of the image is important.

When tracks are not corrupted by a strong noise and when there are no relevant disturbing elements, the AntPX algorithm provides very good results, correctly selecting the path (Fig. 10). For the purpose of assessing the performance in this application, where the snowcats proceed at a low speed, only the results concerning the detection close to the front of the vehicle are relevant, so some discrepancies near the upper side of the analysis field are accepted.

However, there are cases in which the identification of the correct tracks may be a difficult task, due to several problems. For example, as the camera is placed inside the driver's cabin, noise is introduced in the acquisition phase, since the windscreen is often scratched and accentuates possible reflections caused by the sun. Moreover, the terrain structure introduces further difficulties: the various changes in slant cause modifications of the camera view field for uphill and downhill slopes.

[9]Similarly to an existing guiding principle in AS application to TSP, for which the number of ants should equal the number of cities for best performance.

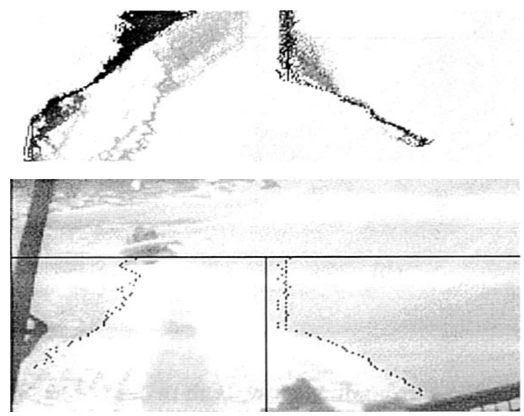
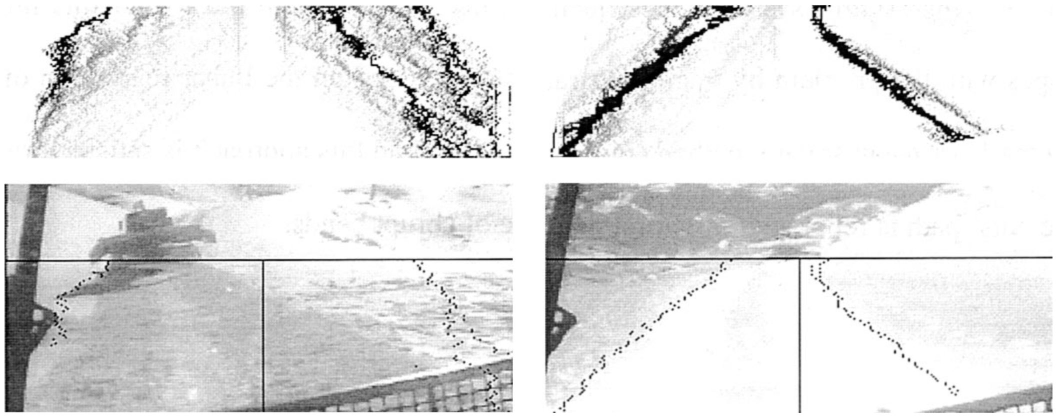


Fig. 11. During a left bend, the right track falls in the left rectangle. In such a case, the ants go on to the top line and AntPX cannot follow the correct track for more than a few meters.

When the vehicle is moving uphill, rectangles of analysis include only a short portion of the tracks, while during downhill movements images contain a longer portion of tracks. Such variations are due to the static nature of the rectangles of analysis, whose size is currently fixed. This causes additional problems when the snowcat takes a bend downhill, as it may happen that a track crosses the central line separating the left and right rectangles. In these cases, the virtual ants go straight on to the top line and assume a gap in the tracks, as they are not allowed to move laterally (Fig. 11).

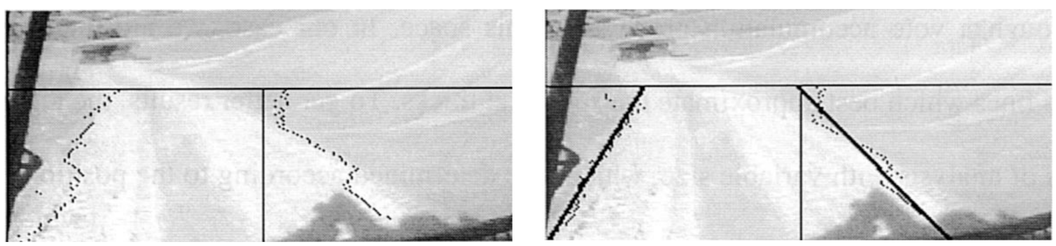
Also, vehicle jolts can cause problems for a precise determination of the probable position of tracks: when many jerks are present, it becomes difficult to identify the correct area to be analyzed, even by a human supervisor.

Other noisy effects found in the test sequences have different origins: for example, marks produced by other vehicles or footprints, the plow and the windscreen wiper, and light and shadow effects. Fig. 12(a), for instance, shows a very critical image in which not only the right marker terminates out of the search area, but it also contains footprints which, connected together, are mistaken for the right track. The same image illustrates an example of failure of the low-level masking process, owing to the relative low contrast of the plow and the snow. Artificial ants



(a) (b)

Fig. 12. (a) Lateral footprints and critical lighting conditions mislead the algorithm. (b) The plow is not considered in the search for tracks.



(a) (b)

Fig. 13. (a) A portion of shadow is mistaken for a track. (b) Tracks approximated by straight lines.

start their run from the bottom line and are mislead by irrelevant information in the area of the plow. On the opposite, in Fig. 12(b) the plow is correctly removed from the area of analysis and the algorithm detects the correct tracks.

Shadows are more difficult to deal with. For instance, in Fig. 13(a) the ants erroneously followed an edge portion of a shadow instead of a track. Using the mask in the low-level phase may not be sufficient to solve this problem, as shadows often have gray level values over the discriminating threshold and are thus accepted as interesting areas.

Unfortunately, the position of the tracks in the proximity of the snowcat is the most relevant piece of data for the driving control system. Consequently, the vision system must infer this information. AntPX copes with the problem by tracing a straight line based on the linear regression of the ants' path using the least mean square method [Fig. 13(b)]. Although this approach is satisfactory when the long-range ants' path is reliable, it might fail in case of abrupt bends.

A. *Comparison With a Hough-Transform-Based Approach*

In order to judge the quality of the proposed algorithm, we compared it with a more traditional method based on the Hough Transform that was previously developed and applied to the same problem of snowcat track detection on the same input data [20].

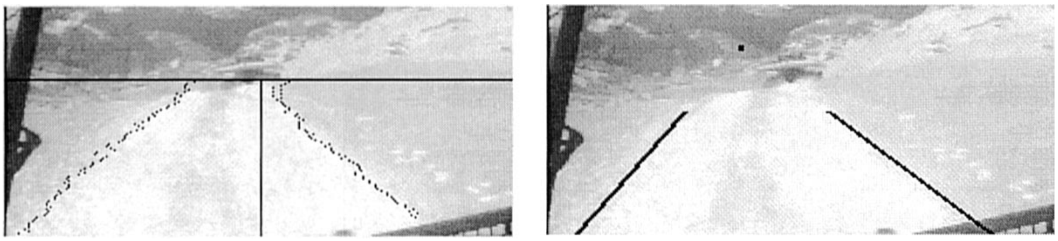
The initial stage of this algorithm is analogous to the one described above and entails a low-level analysis aimed both at extracting the image edges and at masking useless areas (to eliminate the windscreen wiper, the plow, etc.; see Section III). The high-level processing, instead, exploits the Hough Transform principle, which is based on a polar description of straight lines, so that each line has a corresponding point in the polar space. The purpose is to recognize the straight lines present in the image, through a vote accumulation process in this space. In our case, we are interested in finding the two lines which best approximate the snowcat tracks. To get better results, the algorithm uses rectangles of analysis with variable size, which are determined according to the position of the vanishing point (the point resulting from the intersection of the straight lines identifying the tracks in frames immediately preceding the one under analysis). Moreover, a form of correlation between consecutive images is also used, in order to eliminate track estimations that do not maintain a sort of continuity.

Fig. 14 shows the results of the application of the two algorithms, AntPX and the Hough-based one, to the same image. As it is clear, both of them find good approximations for the tracks.

Fig. 15, instead, shows an example where both the methods are confused by a false path and fail in finding the correct track. These are extremely critical situations, where different solutions have a similar strength, and can thus be alternatively selected in different runs of the algorithm on identical or similar subsequent frames; this problem is equally present in both approaches. For instance, Fig. 16 shows accurate results obtained on the frame consecutive to the one in Fig. 15(a).

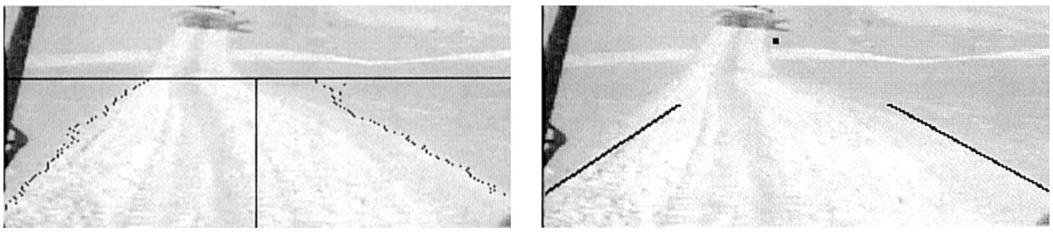
VI. CONCLUSIONS

Extensive and comprehensive tests are needed to accurately interpret the AntPX algorithm behavior in relation to parameters tuning and algorithmic variants; further research is, therefore, required to confirm preliminary experimental settings and



(a) (b)

Fig. 14. Tracks correctly detected with (a) the ant-based algorithm and (b) the Hough-based algorithm.



(a) (b)

Fig. 15. Right track incorrectly found with (a) the ant-based algorithm and (b) the Hough-based algorithm.

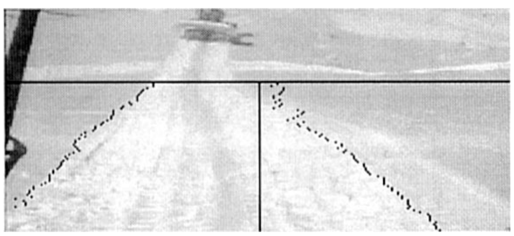


Fig. 16. Right track correctly found with the ant-based algorithm.

to find effective methods for setting optimal parameters values in different situations.

Nevertheless, encouraging experimental results already proved to be potentially competitive against more traditional deterministic approaches and will, thus, stimulate algorithmic improvements and extensions.

In particular, future implementations of the AntPX algorithm will possibly consider alternative pheromone management schemes (such as elitist strategies and pseudorandom-proportional rule variants) and hybrid contributions by complementary techniques and approaches (such as traditional local search algorithms).

Moreover, as previously noted, preceding vehicle detection and vanishing point estimation might help overcome environment-related problems, by allowing resizable rectangles of analysis to adaptively include the actual integral area occupied by the tracks.

Correlation analysis between consecutive frames might additionally smooth undesired effects generated by abrupt terrain-induced vibrations and oscillations.

An image-sensitive mask generation based on variable-threshold might also be introduced within the low-level processing stage to adaptively and effectively remove snowcat-generated noises (shadow, plow and screen-wiper) even under critical lighting situations.

Potential performance improvements might finally derive from a parallel implementation of the AntPX algorithm, both at the data level (left and right track detection) and at the agents' level (ants' activities).

Both approaches (the ant-based and the Hough-based ones) will be tested on site on the South Pole during the next Italian scientific mission in Antarctica during January–March 2002.

ACKNOWLEDGMENT

The authors would like to thank Dr. C. Moriconi of ENEA, Italy, and his institution for the support to their research. Their gratitude goes also to the staff of the ski resort of Passo Tonale (Italian Alps) for their help during the tests. Finally, a special thanks also to L. Fabbri for his preliminary work on the algorithm.

REFERENCES

[1] N. Vandapel, S. Moorehead, W. R. L. Whittaker, R. Chatila, and R. Murrieta.Cid, "Preliminary results on the use of stereo, color cameras and laser sensors in Antarctica," in *Proc. Int. Symp. Experimental Robotics*, Mar. 1999.

[2] A. Foessel, S. Chhedda, and D. Apostolopoulos, "Short-range millimeter-wave radar perception in a polar environment," in *Proc. Field and Service Robotics Conf.*, Aug. 1999.

[3] S. Morehead, R. Simmons, D. Apostolopoulos, and W. R. L. Whittaker, "Autonomous navigation field results of a planetary analog robot in Antarctica," in *Proc. Int. Symp. Artificial Intelligence, Robotics and Automation in Space*.

[4] A. Broggi, M. Bertozzi, and A. Fascioli, "Requirements for visual perception of automotive environments," in *Multimedia Video-Based Surveillance Systems: Requirements, Issues and Solutions*, G. L. Foresti, P. Mahonen, and C. S. Regazzoni, Eds. Norwell, MA: Kluwer, 2000, ch. 1.3.

[5] A. Broggi and E. D. Dickmanns, "Applications of computer vision to intelligent vehicles," *Image Vis. Comput. J.*, vol. 18, no. 5, pp. 365–366, 2000.

[6] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *J. Robot. Auton. Syst.*, vol. 32, no. 1, pp. 1–16, 2000.

[7] A. Broggi, M. Bertozzi, G. Conte, and A. Fascioli, "ARGO prototype vehicle," in *Intelligent Vehicle Technologies*, L. Vlacic, F. Harashima, and M. Parent, Eds. London, U.K.: Butterworth-Heinemann, ch. 14.

[8] A. Broggi, M. Bertozzi, A. Fascioli, and G. Conte, *Automatic Vehicle Guidance: The Experience of the ARGO Autonomous Vehicle*, Singapore: World Scientific, 1999.

[9] M. Dorigo and G. Di Caro, "The ant colony optimization meta-heuristic," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K.: McGraw-Hill, 1999, pp. 11–32.

[10] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the European Conference on Artificial Life (ECAL'91)*, F. Varela and P. Bourgine, Eds. Amsterdam, The Netherlands: Elsevier, 1991, pp. 134–142.

[11] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, *The Travelling Salesman Problem*. Chichester, U.K.: Wiley, 1985.

[12] T. Stützle and H. H. Hoos, "The MAX–MIN ant system and local search for the traveling salesman problem," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, T. Bäck, Z. Michalewicz, and X. Yao, Eds. Piscataway, NJ: IEEE Press, 1997, pp. 309–314.

[13] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," in *Proc. GECCO-2000*, 2000, pp. 893–900.

[14] L. M. Gambardella, E. D. Taillard, and G. Agazzi, "MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K.: McGraw-Hill, 1999, pp. 63–76.

[15] G. Navarro Varela and M. C. Sinclair, "Ant colony optimization for virtual-wavelength-path routing and wavelength allocation," in *Proc. 1999 IEEE Congr. Evolutionary Computation (CEC'99)*, 1999, pp. 1809–1816.

[16] M. Dorigo and T. Stützle, "The ant colony optimization metaheuristic: Algorithms, applications, and advances," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds.. [Online]. Available: http://www.intellektik.informatik.tu-darmstadt.de/~tom/pub.html.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

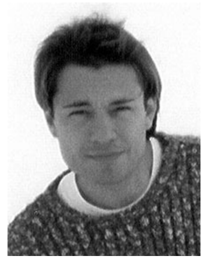
[18] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 53–66, Apr. 1997.

[19] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern.*, pt. B, vol. 26, pp. 29–41, Jan. 1996.

[20] A. Broggi, V. Cantoni, U. Vallone, and A. Fascioli, "Snowcat track detection in snowy environments," in *Proc. 2001 IEEE Intelligent Vehicles Symp. (IV-2001)*, Tokyo, Japan, 2001, pp. 19–24.

Alberto Broggi (S'90–A'96) received the Dr.Eng. degree in electronic engineering and the Ph.D. degree in information technology from the Università di Parma, Parma, Italy, in 1990 and 1994, respectively.

From 1994 to 1998, he was an Associate Researcher in the Dipartimento di Ingegneria dell'Informazione, Università di Parma, where, from 1998 to 2001, he was an Associate Professor of Artificial Intelligence in the Dipartimento di Informatica e Sistemistica, and, since 2001, he has been a Professor of Computer Science. His research interests include real-time computer vision approaches for the navigation of unmanned vehicles, and the development of low-cost computer systems to be used in autonomous agents. He is the Coordinator of the ARGO project, with the aim of designing, developing, and testing the ARGO autonomous prototype vehicle, equipped with special active safety features and enhanced driving capabilities. He has authored more than 120 refereed publications in international journals, book chapters, and conference proceedings. He is actively involved in the organization of scientific events, and is on the Editorial Boards and Program Committees of many international journals and conferences and has been invited to act as Guest Editor of journals and magazines theme issues on topics related to intelligent vehicles, computer vision application, and computer architectures for real-time image processing.



Massimo Cellario received the Dr.Eng. degree in information engineering in 1998 from the University of Pavia, Pavia, Italy, where he is currently working toward the Ph.D. degree in computer science in the Department of Computer and Systems Engineering.

His research interests focus on multimodal/perceptual interfaces for human–computer interaction, 3-D computer vision, and image synthesis.

Paolo Lombardi received the B.E. and M.Sc. degerees in electronic engineering from the University of Pavia, Pavia, Italy, in 2000. He is currently working toward the Ph.D. degree in electronic engineering and computer science within a program of co-tutoring between the University of Pavia and the University of Paris Sud, Paris, France.

His research interests include computer vision, with applications to the automotive industry, and visual attention algorithms.

Marco Porta received the Dr.Eng. degree in electronic engineering from the Politecnico di Milano, Milan, Italy, in 1996, and the Ph.D. degree in electronic and computer engineering from the Università di Pavia, Pavia, Italy, in 1999.

Since January 2000, he has been a Post-Doctoral Researcher at the Università di Pavia. His interests include visual languages and multimedia interfaces and vision-based techniques for human–computer interaction (perceptive user interfaces).