International Journal of Intelligent Control Systems

UNCERTAINTY MODELING AND PERCEPTUAL GUIDING FOR SAFE OPERATION IN UNKNOWN ENVIRONMENTS

M. João Rendas,[1] José Santos-Victor,[2] Jean-Yves Tigli,[1] Isabel Lourtie,[2]
Luc Pronzato,[1] Marie-Claude Thomas,[1] Alexandre Bernardino,[2] Miguel Ribo,[1]
[1] *Laboratoire I3S, CNRS URA 1376, Sophia Antipolis, France*
[2] *Instituto Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal*

This paper addresses the impact of uncertainty modeling and of the use of perceptual guidance on the navigational autonomy for vehicles operating in unknown environments, concentrating around four main topics: security control, planning, perceptual guidance and mapping. We present some of the results obtained in the European Union project NARVAL, which globally aims at using the perception capabilities of an autonomous vehicle to extend its autonomy range.
One of the major issues addressed is related to guarantee vehicle's safety. We propose novel methodologies to *(i)* minimize the risk of loosing the vehicle and *(ii)* avoid destructive interference with objects present in its workspace.
Planning under uncertainty is also studied, and algorithms to define trajectories for guiding the robot to a target region, exploring a delimited environment, or for optimizing autonomous beacon laying, are presented.
The problem of using perceptual data for direct control is discussed in connection with the problem of visual tracking of a moving object.
A new internal knowledge representation structure is proposed, representing the past robot's experience and the acquired knowledge relevant for future planning of the robot's displacements. The problem of maintaining the topological consistency of the internal representation is also considered.

1. Introduction

This paper reports some of the results on the design of control/navigation systems for autonomous mobile robots operating in unknown environments achieved in the European Union Esprit/Ltr project NARVAL.*

The design of systems with a high degree of autonomy is one of the most challenging goals of the scientific community. We distinguish three different kinds of autonomy. **Behavioral autonomy** is concerned with the ability to select and implement the most adequate actions in response to its overall mission context. **Navigational autonomy** is understood as the ability to operate in unknown extended regions. Finally, **temporal autonomy** is related to the possibility of operating continuously over extended periods of time.

Increase of the temporal autonomy is mainly dependent on technological factors. Considerable progress has been made in the last years, with energy sources of large capacity and reduced dimensions.

The first two kinds of autonomy are intimately related. In fact, to "choose and implement the most adequate actions in response to its overall mission context," an autonomous agent must be able to apprehend this "overall mission context". Knowledge about its position and of the surrounding environment – the basic requirements for navigational autonomy – are important components. The NARVAL project concentrates in using the robot's perception capabilities to extend the navigational autonomy of autonomous terrestrial and underwater vehicles.

One of the goals of the project is to show that by a convenient treatment of perceptual information, safe operation in extended unknown spatial regions is possible without resorting to external global positioning information, such as GPS, a priori maps, or specially installed landmarks. This last constraint is particularly important in underwater robotics, where the use of GPS strongly constraints the trajectory of the vehicle, by requiring frequent returns to the ocean surface, and where the installation of landmarks (typically acoustic pingers or transponders) is costly, range limited, and implies an intrusive perturbation of the environment. The project aims at establishing the limits of autonomy of mobile robots in this type of environments, in terms of its sensors observability characteristics and accuracy, the information provided by the environment and, eventually, of the possibility of installing special landmarks for positioning purposes.

We address the impact of uncertainty modeling and of the use of perceptual information on autonomy, concentrating around four main topics: Security Control, Planning, Perceptual Guidance and Mapping.

Under the fundamental constraint of absence (or very reduced availability) of global positioning information, a main concern is on **vehicle security**, in terms of **integrity**, in particular obstacle detection and avoidance, and **autonomy reliability**, meaning that the vehicle must be able to return to a pre-defined homing region.

In *purely metric navigation* of mobile robots, planners generally use geometrical maps of the environment [6] or occupancy grids [16]. *Trajectory planners* use such 2 or 3D representations to find the optimal trajectory

*Navigation of Autonomous Robots Via Active Environment Perception. The project partners are the Centre Nationale de Recherche Scientifique, I3S (F), Instituto Superior Técnico (P), DIST-Università di Genova (I), Thomson-Marconi Sonar (F).

of the robot to reach a goal location (e.g., potential field method [2]). *Path planners* use constrained subspaces of solutions, consisting of waypoints and particular trajectories, leading to different retraction algorithms on a graph structure (e.g., freeways [6], voronoi diagram [24], possible trajectories [20]). These methods are well-adapted for local path planning in static and perfectly known environments with vehicles having a good knowledge of their position. However, such trajectory and path planners present two main drawbacks in the absence of accurate global positionning :

- they do not explicitly consider the possibility of bounding the navigation errors by using particular perceptual behaviors during the robot motion;
- they do not take enough advantage of the complete robot's abilities, such as reactive navigational behaviors, to increase robustness and achieve planning flexibility.

In *purely non-metric navigation* approaches, a plan does not exist or is simply described as a sequence of reactive behaviors that enable the robot to reach a perceptible landmark. In these approaches, an internal representation does not actually exist [7]. The robot can only exhibit reflexive behaviors and wander in search of a particular landmark. This opportunistic way of reaching a goal uses only current perception, without formulating a true plan.

In [1], a human made representation of a priori world knowledge is used. The plan corresponds to a rigid and predefined sequence of reactive behaviors, defining a sequence of landmarks. The robot doesn't produce and increment by itself this high-level representation. This lack of flexibility prevents the robot of using its dynamically acquired knowledge about the world.

Other approaches use tightly coupled perceptual/behavioral representations of the past activity of the robot using graphs (topological map [12], sensorimotor graph [13]). This kind of approaches is well adapted for crowded environments, where the robot can always perceive environmental features. In dispersed environments like underwater areas, and using non sophisticated sensors, these methods do not allow the robot to go through areas lacking perceptual information.

Our methodology combines these two approaches, and is based on the following considerations:

1. The most robust navigation methods drive the robot close to informative environmental features to help reducing positionning errors;
2. A plan is defined in terms of activities, instead of precise trajectories.
3. These activities include not only reactive behaviors but also deliberative activities.

These arguments lead to the internal environment representation proposed, which gathers all the information necessary to define the sequence of actions to achieve a given goal. The representation explicitly describes the confidence of reaching a given target state when a certain command is executed, and it is mainly based on the identification of *gateways* between environment entities. The *gateway* point between two objects defines the best initial conditions to pass in deliberate motion from one object to another. This notion, in some sense dual of the *way-points* of some classical path planners, restricts the search space of the planning system, by guiding the robot to a situation where it uses at best its current knowledge of the morphological characteristics of the objects present in its workspace. Moreover, the gateway points enable an "atomic" approach to the trajectory planning problem, functioning like "reset areas". The behavior of the robot after reaching these areas is dependent only on the local characteristics of the environment and associated uncertainty, and independent of the precise past robot trajectory.

Security control is addressed in Section . To survey the probability that the vehicle is not lost, we develop a stochastic model for the evolution of the pose errors, define the *gateway points*, and compute the uncertainty factors of the global internal representation. Another important aspect of mission security is the ability to detect and avoid obstacles. We present a vision based algorithm that exploits the geometric properties of the vehicle-camera-scene arrangement, and which does not depend on the knowledge of the camera parameters or vehicle motion.

Trajectory planning under uncertain environment representation is addressed in Section . We present an agent-based approach to the problem, and study the definition of trajectories for region survey. Finally, admitting that the robot has the possibility of dropping artificial landmarks at selected positions, the optimization of the localization of self-created reference points is assessed.

Section addresses the problem of Perceptual Guidance, where the use of stereo-video information for tracking a moving target exploiting the non-uniform resolution of log-polar images is discussed.

Mapping is presented in Section , analyzing the problem of updating an internal **uncertain shape representation** of the objects in the environment, and presenting a robust procedure to correct segments of past trajectories done in deliberate motion.

2. Security Control

In this section, two major aspects of mission security are assessed: surveillance of autonomy (section) and obstacle detection and avoidance (section).

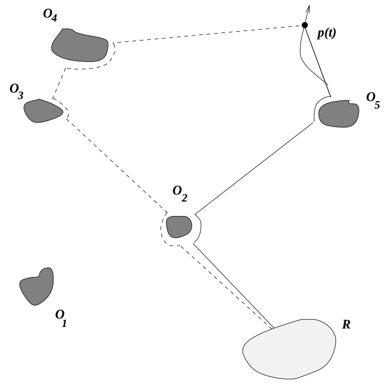


Figure 1: Determination of homing reliability.

2.1. Surveillance of autonomy

Since the growth of positioning errors in an unknown environment strongly depends on the actual events that occur during the course of the mission, vehicle autonomy cannot be determined a priori. To make sure that normal vehicle recovery is achievable at any time instant, an on-board surveillance procedure has been defined that constantly updates the probability, $P_H(t)$, of the vehicle being able to return to a pre-specified homing region, given its current level of positioning uncertainty, the accuracy of the navigation sensors and the learned environment characteristics. Whenever $P_H(t)$ falls below a prespecified level γ, an alarm message is passed to the supervisor unit of the control system, enabling activation of exploratory behaviors or emergency homing routines.

The vehicle's pose (position and attitude) is summarized in the vector $p(t)$, and its current estimate is denoted by $\hat{p}(t)$. To compute $P_H(t)$ we assume that the homing trajectory will be a concatenation of segments of linear uniform motion joining a subset, $O_1, \ldots, O_K$, of the detected objects, see Figure 1, such that

$$\begin{aligned} P_H(t)|_{\text{path } O_1,\ldots,O_K} = \quad & Pr(\text{reaching } O_1 \,|\, \hat{p}(t)).Pr(\text{reaching } O_2 \,|\, \hat{p}_{12}).\cdots \\ & Pr(\text{reaching } O_K \,|\, \hat{p}_{K-1,K}).Pr(\text{reaching } R \,|\, \hat{p}_{KR}), \end{aligned} \tag{2.1}$$

where $\hat{p}_{kl}$ denotes the vehicle's estimate of its pose when leaving object O_k to reach object O_l.

The determination of $P_H(t)$ is based on a judicious choice of the gateway points p_{kl}, which jointly define a set of preferred routess in the set of all possible trajectories. Then, when the vehicle goes from object O_k to object O_l the same gateway point p_{kl} is always used. Our approach, although not optimal from the point of view of path length, is the most adequate for the overall framework considered in the project: autonomous platforms, without global positioning information, moving in an *a priori* unknown environment. It represents a compromise between purely metric navigation and behavioral approaches, in that it tries to take the best profit of the robustness of sensor-based navigation, while admitting the need and possibility of periods of deliberative behaviors.

Each object O_k defines around it a region R_k, the "object's attraction region," where the vehicle is sure to detect the object and acquire the next gateway pose $p_{k,k+1}$. We assume that, given the vehicle's perceptive sensors and the object's characteristics, we are able to determine each object's attraction region.

Using the factors in eq. (2.1) as weights in the links of the graph defined in section , $P_H(t)$ is computed by choosing the path that leads to its maximum value. Under the assumption of linear uniform motion, we obtained an analytical approximation [21] to the growth of the positioning uncertainty over each path segment that enables the fast determination of the link probabilities $Pr(\text{reaching } O_l \,|\, p_{kl})$.

Dynamic and Sensor Models

We present below a general state-space model for the vehicle position and velocity, relating it to the on-board proprioceptive measurements. We distinguish two kinds of sensors: *(i) attitude sensors* (compasses, gyroscopes), providing angular, and/or angular velocity informations defining the attitude of the vehicle with respect to a fixed external frame; *(ii) linear displacement sensors* (accelerometers, velocimeters) providing information about the acceleration or velocity of the vehicle with respect to an internal body frame. In the paper, we consider that the only available sensors are rate sensors (either linear or angular velocities), no conceptual difficulty existing in considering acceleration sensors. Let us define:

- $\underline{x} = [x \; y \; z]^T$, vehicle global position, measured in the external frame.
- $\underline{\dot{x}} = [\dot{x} \; \dot{y} \; \dot{z}]^T$, vehicle linear velocity along each coordinate of the external frame.
- $\bar{v} = [v_x \; v_y \; v_z]^T$, vehicle linear velocity along each coordinate of the body frame.
- $\alpha = [\psi \; \theta \; \phi]$, heading, pitch and roll angles measured in the external frame.

- $\dot{\alpha} = \left[\dot{\psi}\ \dot{\theta}\ \dot{\phi}\right]$, heading, pitch and roll angular velocities measured in the body frame.

The linear velocities in the global frame $\dot{\underline{x}}$ and in the vehicle body frame are related by the attitude angles, $\bar{v} = R_\phi R_\theta R_\psi \dot{\underline{x}}$, where R_ϕ, R_θ and R_ψ are rotation matrices.

The following basic relation holds in continous time :

$$\frac{d}{dt}\begin{bmatrix} \underline{x}(t) \\ \dot{\underline{x}}(t) \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \underline{x}(t) \\ \dot{\underline{x}}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \ddot{\underline{x}}(t) \end{bmatrix}. \tag{2.2}$$

During segments of approximate linear motion, we may consider that the acceleration $\ddot{\underline{x}}(t)$ is a perturbation term modeled as a stationary white Gaussian process. With this model, it can be shown that the periodic observations of the rate sensors, denoted by $z(k)$, are locally described by a linear state-space model of the following general form:

$$\begin{bmatrix} \underline{x}(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} I & F_3 \\ 0 & I \end{bmatrix}\begin{bmatrix} \underline{x}(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} \tag{2.3}$$

$$z(k) = \begin{bmatrix} 0 & H(\alpha) \end{bmatrix}\begin{bmatrix} \underline{x}(k) \\ x_2(k) \end{bmatrix} + v(k)\ , \tag{2.4}$$

where $w_1(\cdot), w_2(\cdot)$ and $v(\cdot)$ are mutually uncorrelated stationary white noise sequences with covariance matrices Q_1, Q_2, and R, respectively. The state component $x_2(\cdot)$ is given by:

$$x_2(k) = \begin{bmatrix} \dot{\underline{x}}(k) \\ \alpha(k) \end{bmatrix} \qquad \text{or} \qquad x_2(k) = \begin{bmatrix} \bar{v}(k) \\ \alpha(k) \end{bmatrix}$$

Matrix F_3 depends on the sampling period, and $H(\alpha)$ is a non-linear function of the vehicle's attitude α. The vehicle characteristics are reflected in the characterization of the dynamical noise components. It is well known that for systems following (2.3)-(2.4), the standard Kalman filter is the optimal MSE estimator.

This model decomposes the state vector into two separate parts, of which only x_2 is observable. For nominal linear motion, $H(\alpha)$ can be taken approximately constant. Next, we present without proof some results concerning the error behavior for systems with this structure. The interested reader is referred to [21].

Error Behavior

For large k, the covariance matrix of the estimation errors of the optimal filter for the system (2.3)-(2.4) has the following structure

$$P(k|k) = \begin{bmatrix} P_1^\infty(0|0) + kC & P_{12_\infty} \\ P_{12_\infty}^T & P_{2_\infty}^f \end{bmatrix}, \tag{2.5}$$

where the "intial condition" for the asymptotic regime of the unobservable component, $P_1^\infty(0|0)$, is a constant, and

$$C = F_3\left(H^T R^{-1} H\right)^{-1} F_3^T + Q_1\,, \qquad P_{12_\infty} = F_3 P_{2_\infty}^f Q_2^{-1} P_{2_\infty}^f\,,$$

are, respectivelly, the rate of growth of its error, and its asymptotic cross-correlation with the error of the observable component. The asymptotic covariance matrix of the filtering estimate of the observable component, $P_{2_\infty}^f$, is the solution of the quadratic equation:

$$\left(P_{2_\infty}^f Q_2^{-1} + I\right) P_{2_\infty}^f = \left(H^T R^{-1} H\right)^{-1}\ . \tag{2.6}$$

The constant $P_1^\infty(0|0)$ depends on the transient regime of the filter, and can only be determined exactly by numerically integrating the Riccati equations. The following analytic approximation to has been obtained [21] by considering, in the Riccati equations, the Kalman gain equal to its steady state value:

$$\begin{aligned} P_1^\infty(0|0) = \ & P_1(0|0) + F_3\left[P_{12}(0|0) - P_{12_\infty}\right]^T \\ & + \left[P_{12}(0|0) - P_{12_\infty}\right] F_3^T + F_3\left[P_2(0|0) - P_{2_\infty}^f\right] F_3^T\ . \end{aligned} \tag{2.7}$$

Using this expression in (2.5) yields an analytical model for the evolution of the covariance matrix of the vehicle position and attitude errors, along segments of uniform motion. This equation shows that the two components of the state vector have distinct asymptotic error behaviors : while $x_2(\cdot)$ tends to a stationary process, with covariance matrix given by $P_{2_\infty}^f$, $\underline{x}(\cdot)$ has a variance that grows linearly with k, a behavior that is characteristic of Brownian motions.

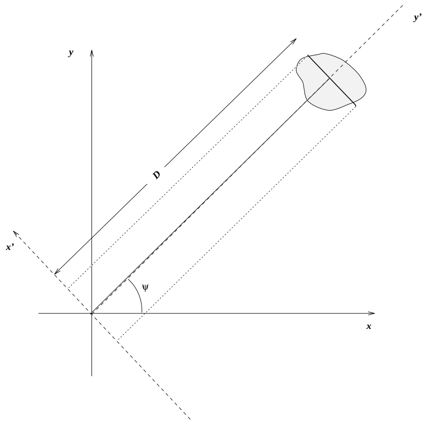


Figure 2: Computation of probability of reaching a distant object (no initial error).

Probability of reaching a given region
The probability, $Pr(\text{reaching } R\,|\,\hat{p}(t))$, of reaching a region R, given a current uncertain position, $\hat{p}(t)$, is given by iterated integrals of the probability density of the vehicle's position, for which no closed form expression is available. We use the Brownian characteristics of the estimation error to compute another probability that approximates the desired one. Figure 2 illustrates our approach. Admit that the initial pose is perfectly known, and that the vehicle is launched in an uniform linear motion from this position along direction ψ, directed at the center of the target. We make a change of variables,

$$\begin{bmatrix} y' \\ x' \end{bmatrix} = R_\psi \begin{bmatrix} x - x(0) \\ y - y(0) \end{bmatrix}, \qquad R_\psi = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix}, \tag{2.8}$$

such that y' coincides with the direction of the linear motion, and the initial position is placed at the origin of the new coordinate system. We use the following approximation

$$Pr(\text{reaching } R\,|\,p(0)) \simeq Pr\left(\forall_{t\in[0,T_R]},\, x'(t) \in [-\ell_R, \ell_R]\right),$$

where $T_R = \frac{D}{v_0}$ is the time required to reach the target with constant nominal velocity v_0, D is the distance from $p(0)$ to the center of the target, and $2\ell_r$ isthe spatial extension of the target in the x' direction. Assimilating $x'(t)$ to a Brownian process, results concerning first passage times and exit times can be used to compute this probability. Let

$$t^*_{\ell_R} = \inf_t \left\{ sup_{0\le s\le t}|x'(s)| \ge \ell_R \right\}$$

be the first exit time from the interval $[-\ell_R, \ell_R]$. Then, it is known [5] that

$$Pr(t^*_{\ell_R} \le \tau) = 2Pr(|x'(\tau)| \ge \ell_R). \tag{2.9}$$

The desired probability can then be written as

$$Pr(\forall_{t\in[0,T_R]},\, x'(t) \in [-\ell_R, \ell_R]) = Pr(t^*_{\ell_R} \ge T_R) = 1 - 2Pr(|x'(T_R)| \ge \ell_R), \tag{2.10}$$

which can be directly computed using the statistical characterization of $x'(\cdot)$ at time T_R.

We consider that, along the nominal trajectory, the estimate coincides with the nominal motion. Then, using $\underline{x}(t) = \hat{\underline{x}}(t) + \epsilon_{\underline{x}}(t)$ and $\hat{\underline{x}}(t) = \underline{x}(0) + v_0[\cos\psi \;\; \sin\psi]^T$, in equation (2.8), we get

$$x'(t) = u_\psi^T \epsilon_{\underline{x}}(t), \qquad u_\psi^T = \begin{bmatrix} -\sin\psi & \cos\psi \end{bmatrix}.$$

With these approximations, $x'(t)$ is a Brownian process, with zero mean and

$$\mathrm{E}\left\{x'(t)^2\right\} = u_\psi^T P_1(t|t) u_\psi.$$

where $P_1(t|t)$ is computed using (2.5).

Until now, we admitted perfectly known initial pose. Uncertainty in initial pose is taken into account by considering that the exit region for computation of the probability (2.10), and the final time T_R, dependend on the actual initial position and attitude errors, $\epsilon_{\underline{x}}, \epsilon_\alpha$, and averaging (2.10) over them. When computating these

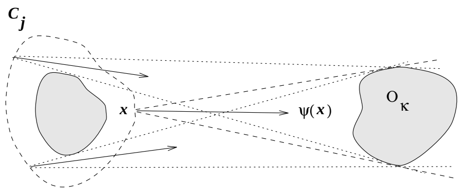


Figure 3: Determination of gateway poses.

averages, we limit the range of values of ϵ_α to those that lead to a nominal trajectory that reaches the target. The probability of reaching a given target is fundamentally dependent on the aperture determined by the target, $\tan(\ell_R/D)$, as it would be expected intuitively.

Identification of gateway poses p_{jk}

The problem of defining poses p_{jk}, i.e., the position and orientation of the robot when leaving O_j to reach O_k, and the associated uncertainty, has been approached in the following way. We consider first the determination of a set $\mathcal{H}_{jk}$ of *remarkable points* (that we will call "handles"), corresponding to positions around object O_j in which the robot's positioning with respect to the object (in position and orientation) when leaving O_j in direction of O_k is locally optimal. We then select the handle that leads to the maximum probability of reaching O_k, computed using the expressions presented previously.

We assume that all the possible launching points from object O_j must lie in the curve

$$\mathcal{C}_j = \left\{ \underline{x} : \min_{x' \in O_j} \|\underline{x} - x'\| = c \right\},$$

i.e., $\mathcal{C}_j$ is the set of points at a fixed distance c from object O_j, see figure 3. Actual determination of c depends on the robot's perception sensors. For 2D robotics, $\mathcal{C}_j$ is a one-dimensional manifold that can be parametrized by arc length ρ, fixing an arbitrary origin : $x = x(\rho)$.

As discussed previously, chosing a point $x \in \mathcal{C}_j$ to attain O_k determines the direction of departure $\psi(x)$, which is always chosen as the direction of the line joining x and the object's center. Therefore, for 2D problems, we can parametrize the set of *admissible gateway poses* $\mathcal{G}_{jk}$ by arc length. The set of *feasible gateway* points $\mathcal{F}_{jk}$ is the subset of $\mathcal{G}_{jk}$ for which the line joining x and the object does not intersect O_j, see figure 3. If all objects have convex shapes, can show that the set of feasible gateway points is the image of a closed interval A:

$$\mathcal{F}_{jk} = \{(\underline{x}(\rho), \psi(\rho)) \,|\rho \in A\}.$$

The set of handles $\mathcal{H}_{jk}$ is the discrete subset of $\mathcal{F}_{jk}$ which collects all the locally "best" poses, the evaluation criterion being based on basic notions of Estimation Theory.

We assume that the estimates of the position and orientation of the robot with respect to O_j are obtained by a Maximum Likelihood processor, i.e., by maximizing the likelihood of the perceptual data z given the robot's pose and the known object characteristics,

$$\hat{p} = arg \max_p p(z \,|\, p, O_j).$$

We can approximate the covariance matrix of the estimation error $\epsilon = p - \hat{p}$ using the Cramer-Rao lower bound :

$$\mathrm{E}\left\{\epsilon\epsilon^T\right\} \simeq \mathrm{CRB}(p) = J(p)^{-1} \tag{2.11}$$

where J is the Fisher information matrix. Using the formula for the inverse of a partitioned matrix, we can write

$$\mathrm{CRB}(\underline{x})^{-1}\big|_\alpha = J(\underline{x}) - J(\underline{x}, \alpha)J(\alpha)^{-1}J(\alpha, \underline{x}).$$

The previous expression shows how the uncertainty of the attitude estimate (described by the block of Fisher information related to α) affects the overall uncertainty of the position estimates. The set of "handle" points is the set of local minima of the determinant of the Cramer-Rao bound for the location $\underline{x}$ over the set of feasible gateway points. For 2D problems,

$$\mathcal{H}_{jk} = \left\{ (\underline{x}(\rho), \psi(\rho)) : \left.\frac{\partial |\, \mathrm{CRB}(\underline{x}(\rho))|_\alpha \,|}{\partial \rho}\right|_{\underline{x}=p(\rho), \alpha=\psi(\rho)} = 0, \rho \in A \right\}.$$

The gateway p_{jk} from object O_j to object O_k is chosen as the "handle" point which yields a larger probability of attaining O_k:

$$p_{jk} = \arg \max_{p \in \mathcal{H}_{jk}} Pr\,(\text{reaching } O_k \,|\, \hat{p}),$$

where $\hat{p}$ is modeled as a Gaussian random variable, with mean p and covariance $J(p)^{-1}$.

When object O_i is added to the internal environment representation, the set of gateway points to all other objects is found and the internal representation of the robot updated. The set of gateways from all other objects to the new object is also identified. The constant update of these "best launching points" defines a set of preferred *routes* throughout the workspace. For identical object's features (i.e., for handles that yield identical uncertainty matrices), the optimization of the error probability is equivalent to chosing the handle that is closest to the target object. However, if the shapes of the objects are not equally informative with respect to the robot's pose, a more distant but also more informative handle can be preferred to a "launching point" that is closer to the target, but which does not allow for orientation acquisition. With our definition, the gateway points correspond to an optimal compromise between knowlegde of position and orientation at the initial point and distance to the target object.

2.2. Obstacle detection and avoidance

The ability to detect and avoid obstacles is an essential capability for a vehicle to operate in an unknown environment. In spite of having poor resolution, ultrasound sensors have been often used instead of vision, since the latter require much more computational power and the robustness achieved, in many systems, is still inadequate.

We consider a vision-based obstacle detection method for a mobile robot, equipped with a single camera, moving over approximately planar structures. The system is robust and suitable for real time implementation, and does not depend on the knowledge of the camera parameters or vehicle motion.

The main idea is the analysis of the particular structure of the optical flow field, when a camera is moving over an approximately planar surface. In Figure 4, Pc is the image plane of a camera moving forwards with pure

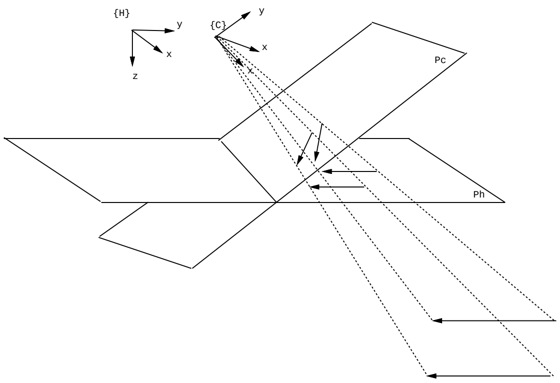


Figure 4: Inverse perspective mapping. The coordinates systems (C) and (H) share the same origin even though in the picture they have been designed separately for simplification. While on (C) the motion of the ground floor is perceived as a complex vectorial pattern, in (H) all the vectors have equal length and orientation under translational motion.

translation, while facing an approximately planar surface. Even in this simple arrangement, the resulting flow pattern is complex, due to the perspective effects, and makes the problem of motion analysis or obstacle detection difficult.

If the camera optical axis were vertical, pointing downwards, the image plane, Ph, would be parallel to the ground plane. In this case, the flow pattern becomes much simpler as the distance to any point on the pavement is constant. All the vectors are equal, and points lying above or below the observed plane can be easily detected. The main idea then, is to inverse project the optical flow in Pc onto the horizontal plane, Ph, as suggested in [15], thus simplifying the original problem.

In [15], the camera intrinsic parameters (focal length, image center, pixel dimension) and extrinsic (camera orientation relative to the ground plane) had to be measured. Instead, we use directly the image measurements to estimate the projective transformation between the image plane and the ground plane. The estimation procedure does not require the calibration of the camera parameters.

Planar surfaces in motion

We consider that the camera observes an approximately planar surface in motion, as the robot moves in the workspace. Assume that the camera has linear velocity $\mathbf{T} = [T_x \ \ T_y \ \ T_z]^T$, and angular velocity $\boldsymbol{\omega} = [\omega_x \ \ \omega_y \ \ \omega_z]^T$. Using the planarity constraint, we can express Z as :

$$Z(X,Y) = Z_0 + \gamma_x X + \gamma_y Y \tag{2.12}$$

where γ_x, γ_y quantify the plane slant and tilt slopes.

The flow of a planar surface in motion is expressed by a second order polynomial in the image coordinates [17], which is valid all over the image. The estimates of the second-order model parameters are often affected by noise which is up to several orders of magnitude higher than the first order parameters estimates, even for perfect planar motion. However, if the angle of view is small and the depth of range limited, the second order parameters can be discarded and the optical flow field of the planar surface approximated by an affine transformation:

$$\begin{aligned} u(x,y) &= u_0 + u_x x + u_y y \\ v(x,y) &= v_0 + v_x x + v_y y \end{aligned} \tag{2.13}$$

where u and v are the x and y components of the flow field; It is assumed that pixel $(0,0)$ is at the image geometric center. The model parameters can be related to the camera motion and to the orientation of the ground plane relative to the image plane:

$$\begin{aligned} u_0 &= -f_x\left[\frac{T_x}{Z_0} - \omega_y\right] & v_0 &= f_y\left[-\frac{T_y}{Z_0} + \omega_x\right] \\ u_x &= \frac{T_z + \gamma_x T_x}{Z_0} & v_x &= \frac{f_y}{f_x}\left[\frac{T_y \gamma_x}{Z_0} - \omega_z\right] \\ u_y &= \frac{f_x}{f_y}\left[\frac{T_x \gamma_y}{Z_0} + \omega_z\right] & v_y &= \frac{T_z + \gamma_y T_y}{Z_0} \end{aligned} \tag{2.14}$$

f_x, f_y denote the camera focal length expressed in pixels (focal length divided by the horizontal and vertical pixel dimensions). The problem that remains to be solved, is the estimation of the plane slant and tilt slopes, which will determine the inverse projection operation.

Plane orientation estimation using the normal flow
The determination of the values of γ_x and γ_y is done in two steps. Fist, the parameters for the affine model are estimated and used, in a second stage, to recover the plane orientation.

The affine motion parameters in equation (2.13) can be directly estimated with 3 measurements of the optical flow $[u \;\; v]^T$. However, using local measurements alone, only the component of the optical flow along the direction of the image gradient (normal flow) can be determined, while the other component remains unknown. This is the well known aperture problem [14]. We rely exclusively on the normal flow field or, equivalently, on the first order time and space image derivatives. Assuming image brightness constancy over time [14], the optical flow constraint equation is given by :

$$uI_x + vI_y = -I_t, \tag{2.15}$$

where I_x, I_y and I_t stand for the partial derivatives of the image with respect to x, y, and t. Combining the optical flow equation with the affine model equations (2.13), we obtain a linear function of the affine flow parameters, $\boldsymbol{\theta}$, that can be estimated from 6 measurements of spatial-temporal image derivatives or 6 measurements of the normal flow. We estimate $\boldsymbol{\theta}$ by using robust estimation methods to discard data outliers.

There are now three different situations to be considered. If the intrinsic parameter are known, γ_x and γ_y can be determined unambiguously from $\boldsymbol{\theta}$. If only the aspect ratio is known, then the parameters can be estimated up to an unknown scale factor, which is sufficient for the method to be applied. Finally, if nothing is known about the intrinsic parameters, we can still use the method, but the robot motion must be constrained to pure translation. Details can be found in [23].

Inverse Perspective Flow Transformation
Similarly to what has been proposed in [15], by inverse projecting the flow field perceived in the image plane, onto the horizontal plane, the resultant flow pattern is much simpler, as illustrated in Figure 4. Let

$$C = \{X_c,\ Y_c,\ Z_c\} \qquad H = \{X_h,\ Y_h,\ Z_h\} \tag{2.16}$$

be the coordinate frames associated to the camera and the horizontal planes, P_c and P_h. These frames differ by a rotation term, ${}^{\mathbf{H}}\mathbf{R}_{\mathbf{C}}$: ${}^{\mathbf{H}}\mathbf{P} = {}^{\mathbf{H}}\mathbf{R}_{\mathbf{C}}\,{}^{\mathbf{C}}\mathbf{P}$, where $\mathbf{P}$ is a point in the 3D space measured in the coordinate frame specified by the upper index. The relative orientation results from a rotation of ψ around the camera x axis, and a rotation of ϕ around the camera y axis (which corresponds to a camera pointing down in front of a mobile robot). The rotation matrix, ${}^{\mathbf{H}}\mathbf{R}_{\mathbf{C}}$, will have the following structure (in most practical systems, ϕ is small and can even be neglected) :

$${}^{\mathbf{H}}\mathbf{R}_{\mathbf{C}} = \begin{bmatrix} c_\phi & -s_\phi s_\psi & -s_\phi c_\psi \\ 0 & c_\psi & -s_\psi \\ s_\phi & c_\phi s_\psi & c_\phi c_\psi \end{bmatrix} \tag{2.17}$$

where $s_\psi, c_\psi, s_\phi, c_\phi$ stand for $\sin\psi$, $\cos\psi$, $\sin\phi$, $\cos\phi$, respectively. The perspective projection of a point in a plane β is defined as:

$$^{\beta}\mathbf{P}' = \mathcal{P}_\beta(^{\beta}\mathbf{P}); \qquad \begin{bmatrix} s\,x'_\beta \\ s\,y'_\beta \\ s \end{bmatrix} = \begin{bmatrix} f_x X_\beta \\ f_y Y_\beta \\ Z_\beta \end{bmatrix} \tag{2.18}$$

where $\mathcal{P}_\beta$ denotes the projection operator, and (x', y') are image coordinates, expressed in pixels. The set of 3D points - $^{\mathbf{C}}\tilde{\mathbf{P}}$ - that project on an image pixel (x'_c, y'_c) defines a beam passing through the projection center and the projection point in the image plane. These points can also be expressed in the frame attached to the horizontal plane:

$$^{\mathbf{H}}\tilde{\mathbf{P}} = {^{\mathbf{H}}\mathbf{R}_{\mathbf{C}}}\,{^{\mathbf{C}}\tilde{\mathbf{P}}}; \qquad ^{\mathbf{C}}\tilde{\mathbf{P}} = [\lambda\frac{x'_c}{f_x} \quad \lambda\frac{y'_c}{f_y} \quad \lambda]^T \tag{2.19}$$

This point can be projected into the horizontal plane, P_h, combining equations (2.17), (2.18):

$$\begin{bmatrix} x'_H \\ \\ y'_H \end{bmatrix} = {^{H}\mathcal{P}_C}(x'_c, y'_c); \qquad \begin{bmatrix} \mu\,x'_H \\ \mu\,y'_H \\ \mu \end{bmatrix} = \begin{bmatrix} c_\phi & -s_\phi s_\psi & -s_\phi c_\psi \\ 0 & c_\psi & -s_\psi \\ s_\phi & c_\phi s_\psi & c_\phi c_\psi \end{bmatrix} \begin{bmatrix} x'_c/f_x \\ y'_c/f_y \\ 1 \end{bmatrix} \tag{2.20}$$

where ${^{H}\mathcal{P}_C}(x'_c, y'_c)$ is a plane-to-plane projective transformation, from the image plane onto the horizontal plane. This equation expresses how to determine the inverse projection in the horizontal plane, of the point projected in the image pixel (x'_c, y'_c). To obtain the inverse projection of a flow vector, the time derivatives of (x'_H, y'_H) must be used:

$$\begin{bmatrix} u'_H(x'_c, y'_c) \\ \\ v'_H(x'_c, y'_c) \end{bmatrix} = \frac{\partial\, {^{H}\mathcal{P}_C}(x'_c, y'_c)}{\partial x'_c}\, u + \frac{\partial\, {^{H}\mathcal{P}_C}(x'_c, y'_c)}{\partial y'_c}\, v$$

The only problem left is the estimation of ψ and ϕ using the affine model and the parameters γ_x, γ_y discussed in the previous sections.

Estimating the Slant and Tilt parameters

During the initialization phase we must determine the inverse projection operator, preferably without explicitly calibrating the system. The process simply consists in determining γ_x, γ_y as explained before. As any point in the camera coordinate system can be expressed in the horizontal plane coordinate system, we can examine the z component of such term:

$$Z_H = s_\phi X + c_\phi s_\psi Y + c_\phi c_\psi Z \tag{2.21}$$

However, in terms of the camera coordinates, Z is a function of X and Y according to the ground plane constraint, given in equation (2.12). Combining these two equations, and knowing that in the coordinate system of the horizontal plane, all the points on the ground surface have a constant depth, Z_H, we get:

$$\psi = -\arctan\gamma_y, \qquad \phi = -\arctan(\gamma_x c_\psi), \qquad Z_H = c_\psi c_\phi\, Z_0. \tag{2.22}$$

Once γ_x and γ_y have been estimated, the optical flow can be inverse projected onto the horizontal plane. The values of ψ, ϕ are estimated only once during the initialization phase and remain constant, provided that the camera is fixed relative to the robot.

Obstacle Detection

In the new coordinate frame, attached to the horizontal plane, the translation is constrained to be parallel to the "*new image*" plane (hence $T_z = 0$), the rotational component is solely around the Z axis and the distance to the ground floor is constant. Therefore, the optical flow is given by :

$$u_h(x_H, y_H) = -\frac{T_x}{Z_H} + \omega_z y_H \qquad v_h(x_H, y_H) = -\frac{T_y}{Z_H} - \omega_z x_H \tag{2.23}$$

There are various situations to consider in the analysis of this flow pattern.

1. $(\omega_z = 0)$ - The optical flow vectors are constant for every point on the ground plane. A simple test can be performed to check if there are any obstacles above or below the ground floor. This detection mechanism relies solely on the fact that the optical flow should be globally constant, no matter what the motion direction and velocity might be. At this point the normal flow vectors can be projected onto the direction of motion, which is constant all over the image.

2. ($T_x = 0$) - In this case, typical of many mobile platforms, u_h can be used to determine ω_z, and used to separate the translation component of v_h. As a result, the remaining flow should become constant for every point lying on the ground floor.

3. If none of the previous cases applies, the main difficulty is how to separate the rotation component from the purely translational component. In our simplified reference frame, one could simply apply an estimation procedure to the inverse projected flow in order to recover the translational (constant all over the image) and rotational components (which depends on the x or y coordinates). The same kind of estimation techniques that were used to estimate the affine flow parameters can again be used to separate the rotation and translation components of motion. Once this has been done, the translational component can be analyzed, as in the former two cases.

Results

The system has been tested in real time, on a mobile robot with a 8mm lens. The camera was placed in the front part of the robot facing the ground plane with an angle of about 65 degrees, with no calibration. The robot speed was set to 10 cm/s. The images resolution is 128×128 pixels and the computations are performed on a central window of 80×80 pixels. The running frequency of the vision loop is around 1Hz.

Figure 5 shows an example of the normal flow field measured during the robot motion, with an obstacle in the field of view. These flow fields were then inverse projected on the horizontal plane, as shown in the middle image. The rightmost image shows the image regions where points lying outside the ground plane, have been found.

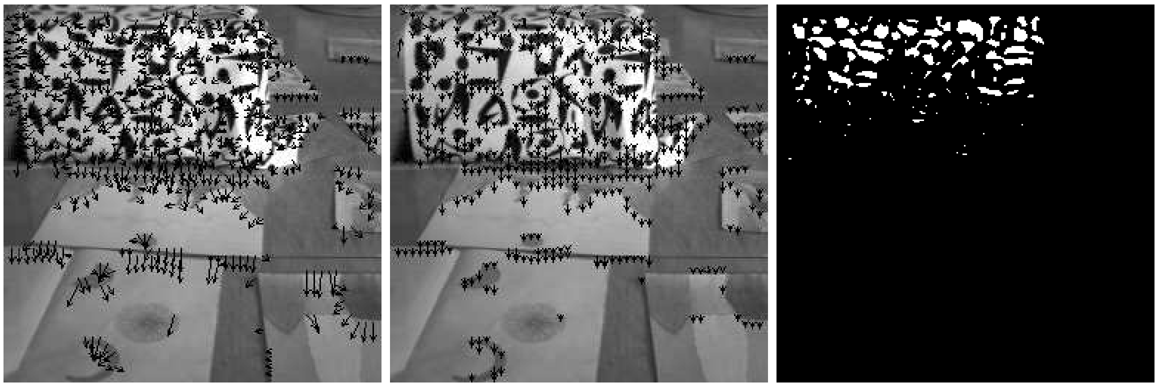


Figure 5: Left: sample of the ground plane normal flow field, during the robot motion. Center: resulting inverse projected flow. Right: detected obstacle.

The resolution of the overall system determines the minimum obstacle size that can be detected, and strongly depends on the image resolution. If more computational power is available, the image resolution can be increased, the same happening with the global system resolution. All these results were obtained without using any information about the camera intrinsic parameters or motion.

3. Planning

The project addressed two distinct problems of trajectory planning under uncertain environment representation. The first problem consists in taking the vehicle from one point to a target region, and has been studied using a distributed agent approach (section). The second one (section) consists in defining a trajectory for exploration of unknown regions.

Finally, we considered the situation where environment information is sparse and mission accomplishment may require the installation of landmarks for future reference. We studied the problem of selecting the placement of these landmarks such that positioning uncertainty is minimized at some target region (section).

3.1. Go to region

In this section, we present a planner that is based on the internal graph representation proposed in section , and that computes the *routing tables* of each node in the graph. Hence, for each possible origin node, we build a list of pairs (goal node, next node) that stores "all pairs shortest paths". According to the underlying goal of achieving a *safe* operation of the robot, we minimize the uncertainty that the designated target node be reached.

Our approach to planning is multi-agent based [9], simulating all the possible evolutions of the robot in the internal graph. Description of the path planning problem consists in designating a pair of nodes. Our algorithm computes the "single node shortest paths" for the current goal node of the robot. It is run separately for each goal node of the graph, defining in this way the complete routing table of each node.

The algorithm for agent propagation is based on their duplication to each neighbouring node, starting from the destination node (first agent). Each agent is represented by its location in the environment graph and by the cost of its path.

This resulting plan representation has the same purpose as the Internalized plans of [19], but applied to a non-metric representation of the environment : "a plan as an advice". It provides an immediate goal-driven information to select the next state of the robot.

3.2. Explore region

This section addresses the problem of determining a trajectory for exploring a closed region delimited by known objects. This problem is motivated by mapping missions. An example, in the framework of underwater robotics, is the exploration of the inner region of an harbour. This problem is distinct from the previous one since it is fundamentally dependent on the "empty part" of the environment, i.e., it must essencially resort to deliberate motions of the robot.

The mapping problem has been studied by several researchers in the last years. Our work differs from the majority of these studies in that we explicitly address the effect of navigation errors in the execution of survey missions. In fact, most of the mapping algorithms that have been proposed consider ideal perfect positioning of the platform along the exploration mission, being based on purely geometric arguments. Usually, this kind of robotic tasks have one of two main objectives: detecting a given object in the environment – e.g. in mine-hunting – and thus ensuring that *all the surface* will be observed; or derive an internal representation of the objects present in the explored region. In either case, blind application of these algorithms to a platform using dead-reckoning navigation will fail to achieve the mission goal: drifting will create "holes" – unobserved areas, increasing the probability of not detecting the objects of interest – and deform the internal representation built with respect to reality.

With these considerations in mind, our approach is based on giving the robot the possibility of knowing exactly where it passed, using available information about the contour of the explored region. The role of perceptual information is to provide global positioning information at the borders of the explored region, and can be either pre-existing (for instance, if the robot started by learning the contour of its workspace) or installed for mission purposes (e.g., a set of landmarks observed using a vision system or a set of short-range acoustical positioning systems).

Since positioning and orientation uncertainty tends to increase most importantly when the vehicle manoeuvers, the exploratory trajectory is assumed to be a sequence of straight line segments. This choice is consistent with the assumption of linear motion when traveling from one object to another that has been used throughout the paper.

A second constraint considers that the location of the extreme points of these line segments must coincide with one of the delimiting objects, allowing in this way repositioning the platform at each of these "contact points".

With these two constraints, the exploratory trajectory e can be defined, for a fixed initial position $p(0)$, by the number of linear segments, n_e, and by the orientation ϕ of each segment with respect to a global frame,

$$e \leftrightarrow (n_e, \{\phi_i(e)\}_{i=1}^{n}).$$

We further assume that the robotic platform has limited range perception sensors, which define an observed region, around each pose of the robot. Our goal is to find a minimal length trajectory which optimizes the use of a priori information, such that the predefined region is completely covered.

We formalise this problem as a multi-criteria optimization problem, considering the following three criteria: *(i)* the lentgh of the trajectory, $L(e)$; *(ii)* the percentage of the complete surface that is effectively observed along trajectory e, $S(e)$; *(iii)* the quality of vehicle self-positioning at the vertices of e, $C(e)$. Our problem consists in finding the number n_{e^*}, and the set of orientations $\phi_i(e^*), i = 1, \ldots, n_{e^*}$, such that $S(e^*) \simeq 1$, $L(e^*)$ is minimized, and $C(e^*)$ is maximized. The first two criteria, $L(e)$ and $S(e)$ have an obvious geometrical definition. The last one was defined using the same performance analysis approach, based on the properties of Maximum Likelihood estimates, which was used to select the handle points in section :

$$C(e) = \sum_{i=1}^{n_t} |\mathrm{CBR}(p_i)|, \tag{3.24}$$

where p_i is the pose at the end of the i-th trajectory segment.

The criteria $S(e)$ and $C(e)$ are highly non-linear functions, leading to a non-linear multi-variate multi-criteria problem. Additionally, the dimension of the solution, i.e., the number of linear segments needed to cover the complete surface, is not known. In general, there is not *a single optimal* solution, as there may exist several distinct trajectories that lead to the same value of the three criteria. Standard optimization techniques cannot deal with these problems. We applied stochastic optimization tools (genetic algorithms) in order to determine a *set of trajectories* which correspond to distinct trade-offs between the three criteria involved.

In the first phase of this work, only the requirements of minimal length and complete coverage, which allow an easier interpretation and analysis of the solutions found, have been considered [26]. Incorporation of the criteria measuring uncertainty, using (3.24), presents no conceptual difficulty, implying mainly an increased computational effort.

Considering each possible trajectory as an individual, genetic mutation and reproduction operators were defined. Three different *mutation* operators were designed: elimination of a randomly chosen linear segment; introduction of a randomly chosen segment at a random point; and random modification of a randomly chosen

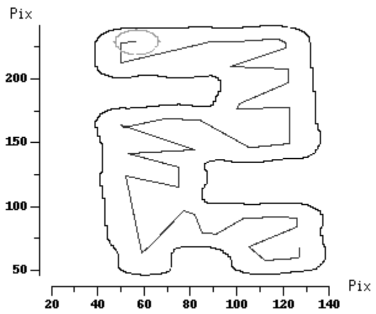


Figure 6: Exploratory trajectory.

departure direction. The *reproduction* operator (applied to trajectory pairs) randomly selects two points (one in each parent) and builds a new pair of trajectories by joining the original parent trajectories at the selected points.

Two approaches to the definition of the selection mechanism for this multi-criteria problem have been studied. The first is based on the total ordering of the population with respect to a single criteria which algebraically combines fittness values for each separate criteria, and is seen to rapidly lead to strong genetic pressure, leading to homogeneous populations that correspond to one possible trade-off between all criteria. The second approach uses a selection mechanism that assumes only partial ordering of the population, and is based on the notion of Pareto set [11]. This approach has the advantage of identifying a set of possible solutions, that correspond to different compromises between the criteria considered. Our experiments show that the genetic algorithms are able to "discover" regular geometric solutions, using for instance, sets of parallel lines, diagonals, etc, see figure 6, where an example of one trajectory generated by the algorithm is presented.

Current work integrates the uncertainty criterion in the algorithm, enabling the definition of exploratory trajectories that are "supported" by perceptual knowledge, and whose geometry depends on the local perceptual features of the contour surface, as well as the global geometrical properties of the surface to be covered.

3.3. Beacon laying

Long-range missions may drive the robot far away from known beacons B_i, $i = 1, \ldots, n_B$, and may thus imply a deterioration of the precision on its estimated position. In this case, it might be profitable to allow the robot to lay down its own beacons B_i', $i = 1, \ldots, n_{B'}$, at some particular positions to be optimized. Noisy measurements of distances to the B_i''s are then used to improve precision on the robot position, which permits to extend its autonomy.

Assume that the robot drops a beacon B' at time k, so that at future time instants $j > k$ the distance to B' will be observed. Extended Kalman Filtering (EKF) can be used to process this information, taking into account the uncertainty on the location of B', due to uncertain robot position when laying the beacon down. For that purpose we suggest to extend the state of the system, to include the (x, y) position of the beacon. At time $j \geq k$ we thus have an extended state $\boldsymbol{x}^e$,

$$\boldsymbol{x}_j^e = \left[\begin{array}{ccc} \boldsymbol{x}_j^T & x_j^{B'} & y_j^{B'} \end{array} \right]^T$$

with $(x_j^{B'}, y_j^{B'})$ being the coordinates of the beacon and $\boldsymbol{x}_j$ the original state vector corresponding to the dynamics of the vehicle. The time evolution of $\boldsymbol{x}_j$ is described as

$$\boldsymbol{x}_{j+1} = \boldsymbol{f}(\boldsymbol{x}_j, \boldsymbol{u}_j) + \boldsymbol{v}_j \ ,$$

where $\boldsymbol{u}_j$ is the control applied at time j and $\boldsymbol{v}_j$ a corrupting noise, assumed to belong to an i.i.d. sequence, with $E\{\boldsymbol{v}_j\} = \boldsymbol{0}$ and $E\{\boldsymbol{v}_k \boldsymbol{v}_l\} = \delta_{kl} \boldsymbol{V}_k$. Since B' is fixed, the evolution equation for $\boldsymbol{x}_j^e$ is simply

$$\boldsymbol{x}_{j+1}^e = \left[\begin{array}{c} \boldsymbol{f}(\boldsymbol{x}_j, \boldsymbol{u}_j) \\ x_j^{B'} \\ y_j^{B'} \end{array} \right] + \boldsymbol{v}_j^e \ , \quad \text{with} \quad E\{\boldsymbol{v}_j^e \boldsymbol{v}_l^e\} = \delta_{kl} \left[\begin{array}{cc} \boldsymbol{V}_j & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{O} \end{array} \right] . \tag{3.25}$$

When EKF is used to estimate $\boldsymbol{x}_j^e$, $j \geq k$, the last two components of $\hat{\boldsymbol{x}}_{j|j}^e$ are initialized at time k as $\hat{x}_{k|k}^{B'} = \hat{x}_{k|k}$, $\hat{y}_{k|k}^{B'} = \hat{y}_{k|k}$, with $(\hat{x}_{k|k}, \hat{y}_{k|k})$ the estimated position of the robot at time k. The associated covariance matrix $\boldsymbol{P}_{j|j}^e$

is initialized as

$$\boldsymbol{P}^e_{k|k} = \begin{bmatrix} \boldsymbol{P}^1_{k|k} & \boldsymbol{P}^2_{k|k} & \boldsymbol{P}^1_{k|k} \\ \boldsymbol{P}^{2^T}_{k|k} & \boldsymbol{P}^3_{k|k} & \boldsymbol{P}^{2^T}_{k|k} \\ \boldsymbol{P}^1_{k|k} & \boldsymbol{P}^2_{k|k} & \boldsymbol{P}^1_{k|k} \end{bmatrix},$$

with $\boldsymbol{P}^1_{k|k}$ the covariance matrix associated with the components $(\hat{x}_{k|k}, \hat{y}_{k|k})$ of $\hat{\boldsymbol{x}}_{k|k}$ and $\boldsymbol{P}^2_{k|k}$, $\boldsymbol{P}^3_{k|k}$ obtained from the following partition of $\boldsymbol{P}_k$:

$$\boldsymbol{P}_{k|k} = \begin{bmatrix} \boldsymbol{P}^1_{k|k} & \boldsymbol{P}^2_{k|k} \\ \boldsymbol{P}^{2^T}_{k|k} & \boldsymbol{P}^3_{k|k} \end{bmatrix}.$$

The state extension procedure can be repeated each time a new beacon is laid down.

The components $(\hat{x}^{B'}_{j|j}, \hat{y}^{B'}_{j|j})$ and the associated part $\boldsymbol{P}^{B'}_{j|j}$ of the covariance matrix $\boldsymbol{P}^e_{j|j}$ remain constant if B' is the unique beacon in the environment of the robot. An interesting feature of this approach is that $(\hat{x}^{B'}_{j|j}, \hat{y}^{B'}_{j|j})$ and $\boldsymbol{P}^{B'}_{j|j}$ evolve with j when other beacons are present. Note that $\boldsymbol{P}^{B'}_{j|j}$ can only decrease due to the assumed evolution equation for $(x^{B'}_j, y^{B'}_j)$, see (3.25).

Example:
Consider a three-wheeled vehicle (Khepera robot), with dynamics described by the model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \frac{r}{2} \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2/E & -2/E \end{bmatrix} \begin{bmatrix} \omega_r(t) \\ \omega_l(t) \end{bmatrix},$$

where $E = 5.3$cm is the distance between the two front wheels and $r = 0.75$cm is their radius, ω_r (resp. ω_l) denotes the angular speed of the right (resp. left) wheel. They are assumed to satisfy the constraints $|\omega_r| \leq \omega_0$, $|\omega_l| \leq \omega_0$, $\omega_0 = 26.17$rad/s. An approximated discretized model is used for EKF, given by

$$\boldsymbol{f}(\boldsymbol{x}_k) = \boldsymbol{x}_k + T\frac{r}{2} \begin{bmatrix} \cos\theta_k & \cos\theta_k \\ \sin\theta_k & \sin\theta_k \\ 2/E & -2/E \end{bmatrix} \boldsymbol{u}_k,$$

with $T = 0.05$s,

$$\boldsymbol{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x(kT) \\ y(kT) \\ \theta(kT) \end{bmatrix}, \quad \boldsymbol{u}_k = \begin{bmatrix} \omega_r(kT) \\ \omega_l(kT) \end{bmatrix}.$$

The observation equation is given by

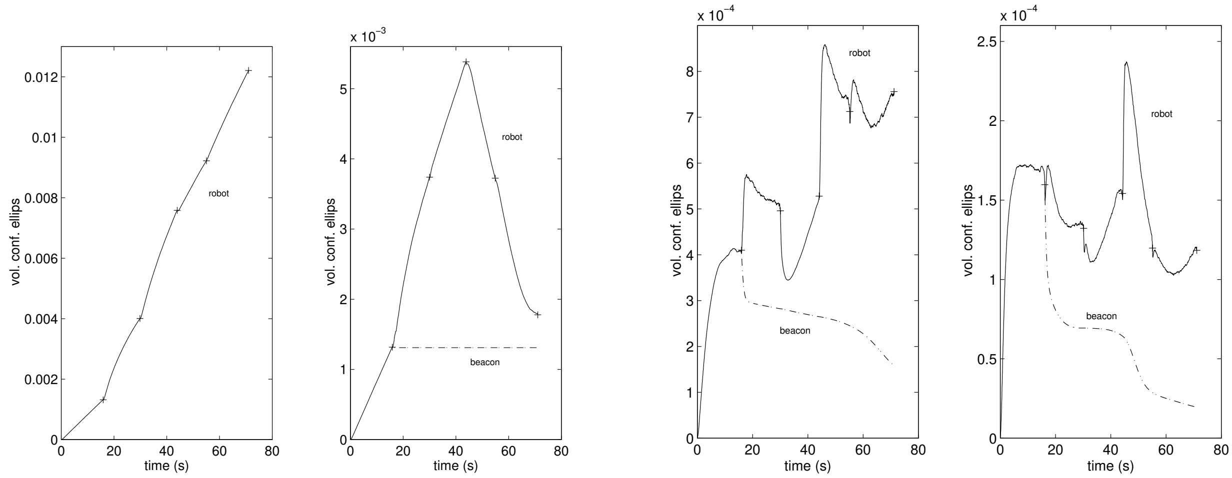
$$\boldsymbol{z}_k = \begin{bmatrix} \theta_k \\ ((x_k - x^{B_1})^2 + (y_k - y^{B_1})^2)^{1/2} \\ \vdots \\ ((x_k - x^{B'_{n_{B'}}})^2 + (y_k - y^{B'_{n_{B'}}})^2)^{1/2} \end{bmatrix} + \boldsymbol{w}_k$$

with (x^{B_i}, y^{B_i}) the coordinates of the beacon B_i ($z_k = \theta_k + w_k$ if no beacon is present). The noise covariance matrices are as follows:

$$\begin{aligned} \boldsymbol{V}_k &= \mathrm{diag}(10^{-6}, 10^{-6}, 10^{-3}), \\ \boldsymbol{W}_k &= \mathrm{diag}(10^{-2}, 10^{-4}d_1^2, \ldots, 10^{-4}d^2_{n_B+n_{B'}}), \end{aligned}$$

where d_i is the distance to the ith beacon (measured distances are used for simulating the system, estimated values are used in EKF). A nonlinear one-step-ahead predictive controller [22] is used to drive the robot towards fixed targets T_i, $i = 1, \ldots, n_T$.

Figure 7a presents the time evolution of the volume of confidence ellipses for the position of the robot, starting at $(0,0)$, along its path through the targets $(4,1), (6,4), (3,6), (1,0)$ and $(0,0)$. A cross is plotted each time a target is reached. No beacon is present, and uncertainty increases monotonously. Figure 7b concerns the case where a beacon B' is dropped at $(4,1)$. The volume of confidence ellipses on $(x^{B'}, y^{B'})$ is plotted in dashed line: the precision on the position of the robot is improved, that on the position of B' remains constant. The case in Figure 7c (resp. 7d) is similar to that in Figure 7b, except that a known beacon B is present in $(10,0)$ (resp. two



(a) (b) (c) (d)

Figure 7: Volume of confidence ellipses versus time. (a) no beacon is present; (b) one beacon dropped; (c) one known beacon and one beacon dropped; (d) two known beacons and one beacon dropped.

known beacons at $(10, 0)$ and $(0, 10)$). The precision on the position of B' now improves with time. The optimal dropping site for B' is still to be determined. This is done in the next section.

Optimal choice of beacon laying

We consider the same robot as in the example above. The mission is defined as follows: starting from $(0, 0)$, the robot should visit the targets T_1, T_2 (in this order), respectively at $(8, 4)$ and $(3, 10)$, and reach T_2 with maximal precision. A beacon B' can be laid down at any place during the mission. Assume that a known beacon is present at (5,5). Figure 8 presents the level sets of the volume of confidence ellipses on the robot position when it reaches T_2 (obtained by noise-free simulation) as a function of the location of B', dropped *before* visiting T_1 (the case where B' is dropped *after* visiting T_1 must also be considered). The optimal strategy is: the robot first goes to T_2, drops B', then goes to T_1 and finally returns to T_2.

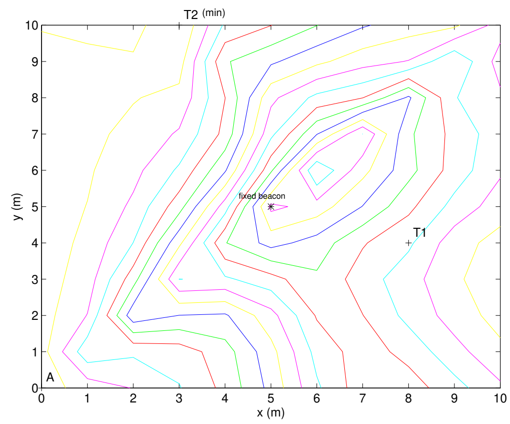


Figure 8: Level sets for the volume of confidence ellipses for the position of the robot when it reaches T_2. $+$: targets, $*$: beacon.

Exchange algorithms [10] can be used when more beacons can be laid down. In particular, the proposed approach may be of special interest when distance to known and dropped beacons can be measured only in a limited range (e.g. when the variance of the measurement noise quickly increases with the distance). In the example, the robot has to visit a given set of targets, and the trajectory is chosen so as to maximize the precision on its location when the last target is reached. Other criteria, taking into account the precision for the other targets and/or the total length of the robot trajectory, could be used as well.

4. Perceptual Guidance: Tracking a moving target

The ability to track a moving target is one of the goals to be developed within the project. For this purpose, we will use a stereo head - Medusa - with four degrees of freedom : independent vergence, tilt and pan, shown in Figure 9.

In nature, one of the main reasons for the existence of active head-eye control is the space-variant density of photo-receptors in the eye, distinguishing between a central high resolution area - the fovea, and the low resolution peripheral zone.

The non-uniform resolution of the human retina has been found to follow a logarithmic-polar law, taking into account the linear increment of the photo-receptors size with the distance to the center [25]. Combining this feature with the ability to move the eyes over the visual field, it is possible to obtain the information needed to perform the desired tasks, without considering all the data contained in uniformly sampled images. Hence, the processing effort can be concentrated in the foveal area.

In practice, one can show that log-polar images have various advantages over the cartesian images, like shape invariance to scaling and rotation, improved binocular fusion and adequate correlation properties to vergence control [3].

Log-Polar Mapping

The main reason for ocular movements in many biological systems is the existence of a space variant resolution in the retina. In humans, the receptors at the retina are distributed in space with increasing density toward the center of the visual field (the fovea). Therefore, eye movements are used by the human oculomotor system to point the central high resolution fovea in the direction of the scene areas to observe more carefully.

Apart from few exceptions [18], most active vision systems use cartesian image representations. Instead, in the vergence system described here is based on a log-polar space variant image sampling. The main advantage of this kind of sampling is an important data reduction obtained at the cost of reducing the resolution in the periphery of the images, but the high resolution in the fovea is sufficient to accomplish most of the visual tasks.

The log-polar transformation is a conformal mapping from the points on the cartesian plane (x, y) to points in the log-polar plane (ξ, η), [25]. The mapping is described by :

$$\begin{cases} \xi & = \log_k(\rho) \\ \eta & = \theta \end{cases} \tag{4.26}$$

where k represents a decaying resolution factor and ρ and θ denote the commonly used polar coordinates. Figure 9 shows the discretized cartesian and log-polar planes.

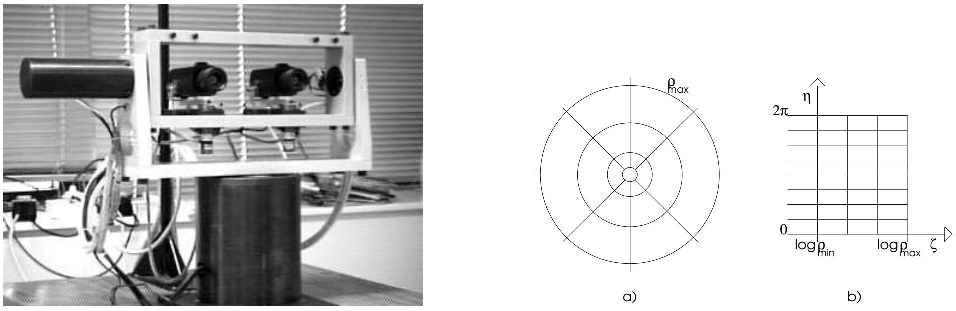


Figure 9: Left: Medusa stereo head used for the experimental tests. Right: Space variant mapping from the cartesian plane, (a), to the log-polar plane, (b).

Correlation-based Vergence

When following a target, the motion of the human eyes can be decomposed into saccadic, smooth pursuit and vergence movements. Saccadic and smooth pursuit movements use image position and velocity errors, in order to compensate lateral motion of the target. Vergence movements, instead, are based on disparity cues to compensate motion in depth.

When observing a target, moving away along the same gaze direction, our eyes rotate by equal amounts in opposite directions. The purpose of the vergence movements is to control the angle between the optic axes, in such a way that a fixation point along some specified gaze direction is kept at the center of the visual field.

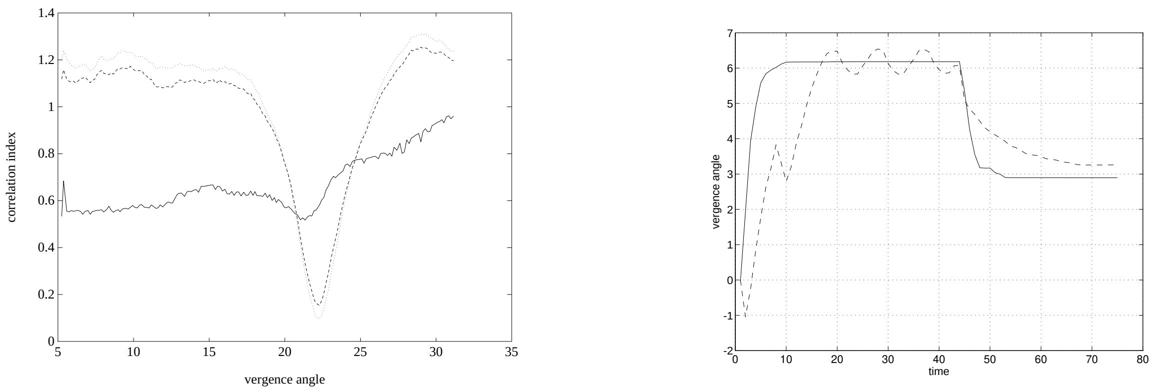
In algorithm detailed here, we use correlation measures between log-polar images to achieve vergence. Our vergence control strategy is based on a set of n channels, tuned to a set of n horizontal disparities $\mathcal{D} = \{d_1, \cdots, d_n\}$, defined on the cartesian plane.

For each channel with disparity d_i, we determine the corresponding disparity map in the log-polar plane. This procedure can be done off-line. Then, we compute the correlation of the left image with the right image, after warping by the log-polar disparity map corresponding to each channel. The correlation value indicates the likelihood of the images having an horizontal cartesian disparity d_i. Therefore, the channel yielding the highest correlation corresponds to the dominant image disparity. As disparity changes, control switches to other channel, resulting in a sequence of pre-programmed responses. In the current implementation, vergence is achieved at video-rate (25Hz) on a 133MHz Pentium computer.

We have performed experiments to illustrate the advantages of the log-polar geometry over the cartesian images, for achieving vergence. An object was placed in front of the camera rig and the cameras moved to sweep the full vergence space. The log-polar images are 8 times smaller then the original cartesian images. Figure 10a shows a plot of the correlation index, as a function of the vergence angle, both for the cartesian and log-polar images.

This experiment shows that for the cartesian images, the correlation index exhibits various local minima and a nonuniform behavior. On the other hand, for the log-polar images, with 8 times less data, the curve is much smoother and presents a sharp minimum in the vicinity of the correct vergence angle.

Figure 10b shows the time evolution of the vergence angle for an experiment with a target moving back and forth in front of the head. The time scale units are sampling periods (about 40ms).



(a) (b)

Figure 10: **(a)** $\mathcal{C}(\nu)$: correlation index as a function of the vergence angle. Solid line: $\mathcal{C}(\nu)$ for a cartesian pair of images. Dashed and dotted lines: $\mathcal{C}(\nu)$ for log-polar images with different layouts. The minimum values represent the correct vergence angle. **(b)** Closed loop response to step-like stimuli (solid plot).

Smooth pursuit

The vergence process enables foveated animals to register objects of interest on the fovea of each eye. As a consequence, it allows the extraction of the greatest possible amount of information. Moreover, when verging on a target, it is possible to find its retinotopic location through disparity filtering, thus allowing to isolate the target from the background [8].

Once the target is segmented in the image, we can use information about its velocity to implement the *smooth pursuit* mechanism that compensates for the target lateral motion.

We use the information about the target normal flow (refer to section for a discussion on the optical flow) to estimate the target position in the image and its velocity. These position/velocity measurements are used to drive the motion of the cameras, to compensate the target motion. This is the so-called *smooth-pursuit* movement. Currently, vergence *and* tracking are integrated at half video-rate - 12.5Hz.

5. Mapping

We describe in this section a novel approach to the creation and update of an uncertain workspace representation for vehicles equipped of inertial and perception sensors. We present in section the structure of the internal representation that is built by the robot, and that collects all information relevant for the definition of safe plans. We give also a short report on studies concerning trajectory correction by integration of independent global position and attitude information, for which we developed a robust sub-optimal procedure (section).

5.1. Internal representation

This section describes the internal representation which is dynamically updated by the robot, which summarizes the relevant robot's past experience and acquired knowledge. As discussed in section , this representation is motivated by the goal of accepting the need of periods of dead-reckoning navigation, and of taking the best possible advantage from the information provided by the environment.

This representation is based on the notion of "distinguishable states", that we define as the atomic global state of the robot, collecting its perceptual, behavioral and internal conditions. The internal knowledge is organized as a dynamic directed graph $G(V, E)$, where V is the set of nodes and E is the set of edges.

Definition of Distinguishable state and Transition
A distinguishable state of the robot is a triplet, describing not only its behavioral and perceptual conditions but also its internal state. The robot *perceptual state* uses the notion of "landmark" as a high level codification of the the robot's perceptual data. A general landmark descriptor is a tuple (T, P, G, I), where T is the type (e.g. circular or linear objects), P is the absolute position, G are geometrical parameters and I uncertainty coefficients. This data is not used directly by the planner (see section) but is required to compute the uncertainty about reaching one given state from another, and correctly determine the current sensorimotor state of the robot.

The robot *behavioral state* can describe all the activities of the robot. We distinguish reactive, semi-deliberative and deliberative navigational behaviors. The *internal state* of the robot represents the localization of the robot and the associated uncertainty. The *transitions* between distinguishable states are represented by the edges of the graph. Each edge is labeled by the uncertainty in the effective achievement of the target state.

We note that the following main extensions to the classical sensorimotor approaches are proposed :

- Distinguishable states contain an uncertain description of the internal state of the robot: This is useful to explicitly restrict the chaining of robot's activities, assuring that a high enough level of confidence is achieved before attempting critical manoeuvers.
- Possible behaviors include semi-deliberative and deliberative actions. This is a basic requirement to achieve non-oportunistic, efficient behaviors of the robot in dispersed environments, increasing the robot's effective workspace beyond the immediate neighborhood of the connected set of objects close to its initial position. The precise notion of connectedness depends on the range of the perceptual sensors.
- Transitions are attached a level of uncertainty, allowing the consideration of the possibility of error recovery during planning. The exact definition of these labels has been addressed in section , and is fundamentally based in the notion of "gateway" points.

Example
A possible robot's environment is shown in figure 11, together with the resulting graph knowledge representation. In this example, we consider the following set of possible robot's actions:

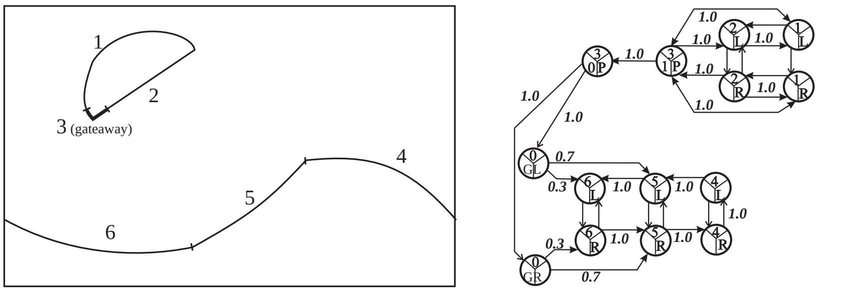


Figure 11: Left: Example of environment. Right: graph using gateaway states.

- *Reactive behaviors:* following landmarks on the left (L) or on the right (R);
- *Semi-deliberative behaviors:* manoeuvring for positionning in a particular gateway (P).
- *Deliberative behaviors:* following a predefined trajectory until a landmark is detected and then following it on the left (GL) or on the right (GR).

The probability of reaching landmark 2 when the robot is in distinguishable state (1,R) is 1, but the probability of reaching landmark 6 in the distinguishable state (0,GR) is much smaller, being equal to 0.3. In fact, the robot can reach landmark 6 but also landmark 5. The probability of reaching landmark 5 or landmark 6 are described by the uncertainty coefficients that label the graph's edges: 0.7 for landmark 5 and 0.3 for landmark 6. Note that our internal representation is totally connected, allowing the robot to go through an a priori non-limited environment.

To reach the gateway area 3 the robot uses a manoeuvering behavior (semi-deliberative). Two nodes are used in the graph to represent the departure from gateway 3: nodes (3,P,1) and (3,P,0). When the robot is manoeuvering to reach gateway area 3 but not yet with enough precision, it is in distinguishable state (3,P,1). When the robot gains enough precision in its location with respect to gateway 3, it passes to state (3,P,0). In this case, the transition corresponds to a change in internal state.

5.2. Trajectory reconstruction

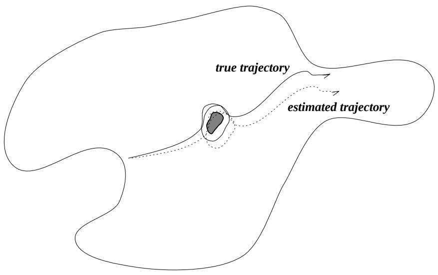


Figure 12: Correction of past trajectories.

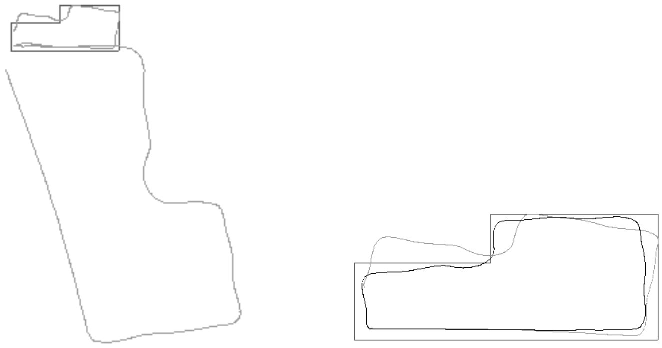


Figure 13: (a) optimal linear smoother; (b) two-step method.

We considered the problem of correction of trajectory segments previously estimated by dead-reckoning when external global position information is made available, see figure 12. This problem is important to reposition newly detected objects with respect to other elements already existing in the map, maximizing the global spatial coherency of the representation. It is fundamental, for instance, in the context of surveying missions, to be able to correctly positioning objects found inside the external contour (see section).

Three sub-optimal solutions have been defined and compared [4]. One consists in the application of the optimal smoother to a linearized model of the system, while the other two are two-step procedures, where a new trajectory estimate is built backwards in a first step (using as initial condition the new pose information), and then pointwise combined, in a second step, with the initial estimate (obtained in the forward direction). These last two methods differ on the data they process: while one assumes that the inertial data has been recorded during dead-reckoning navigation, the second re-synthesizes inertial data using the forward trajectory estimate. These methods have been tested in a mini-platform equipped of odometric sensors, the last two approaches offering a better performance that straightforward use of the (extended) optimal smoother, as it is shown in figure 13 where the initial (gray line) and corrected (dark line) trajectories are shown for the first and second methods. In this experiment, the robot followed a trajectory approximately parallel to the closed polygonal contour shown. When it reached its initial position, it was manually stopped. The corrected trajectories were obtained by using the algorithms with initial conditions obtained by direct (manual) observation of the robot's final position and orientation. As it can be seen, the use of a (wrong) linearized model to constrain the complete trajectory of the robot leads to a completely distorted reconstructed trajectory, while the other method performs well, even if, as in this example, a large correction of the final position and attitude are required.

6. Conclusions

The paper discussed several problems concerning the definition of autonomous systems able to navigate in unknown extended environments. We recall here the major issues addressed and the approaches underlying the solutions proposed.

One of the major contributions concerns the definition of an uncertain representation that summarizes all knowledge useful for planning future motions of the robot, and that allows continuous survey of its autonomy. This internal world model explicitly identifies the environmental features that can be of most help for navigation purposes. The approach used to define the basic elements of this representation is strongly grounded on statistical analysis tools. It allows an approach to the path planning problem which combines characteristics of purely metric and behavioral methods, taking advantages of the efficiency and generality of the first, and inheriting the robustness properties of the later. Several related issues of more detailed nature were also addressed – autonomous selection of artificial reference points, consistency of the spatial (topological) relations amongst the elements of the internal representation, generation of exploratory trajectories – under the overall high uncertainty conditions assumed throughout the paper.

We have also presented two vision-based behaviors which can extend the autonomy of the vehicle. The first one adds the possibility of tracking a moving target. One of its most salient features is the use of space variant image and the control strategies. The second behavior allows the detection (and avoidance) of obstacles lying above an approximately planar surface. These behaviors highlight the intimate relationship between perception and control of action, yielding a robust performance.

The set of contributions presented here establishes the ground for the ultimate goal set out in the introduction of the paper: defining an exploratory behavior which makes full use of the robot's perceptual capabilities in order to extend its autonomy range, enabling it to perform missions in unknown distant regions without getting lost, by actively searching and identifying the environmental features that are necessary to safely support its progression throughout its workspace.

Acknowledgments

This work has been partially funded by the European Union through the ESPRIT (Long Term Research) project no. 20185. The authors want to acknowledge the collaboration of Wilfrid Têtenoire and Fabrice Bouvier in the development of the software and the execution of the experiments reported in sections and .

References

1. R.C. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6(1-2):105–122, June 1990.
2. J. Barraquand, J.C. Latombe. Nonholonomic multibody mobile robots : Controllability and motion planning in the presence of obstacles. In *Proc. of The 1991 IEEE Int. Conf. on Robotics and Automation*, pp 2328–2335, April 1991.
3. A. Bernardino, J. Santos-Victor. Vergence Control for Robotic Heads using Log-polar Images, IEEE/RSJ Int. Conf. on Intel. Robots and Systems - IROS96, Japan, Nov. 1996,
4. F. Bouvier, M.J. Rendas. Estimation et correction statistique de trajectoires en robotique mobile par intégration d'informations globales. *CNRS, Lab. I3S*, (96-44), Sep. 1996.
5. L. Breiman. Probability. Classics in Applied Mathematics Series, SIAM, 1992.
6. R.A. Brooks. Solving the find-path problem by good representation of free space. In I.J. Cox, G.T. Wilfong, eds, *Autonomous Robot Vehicles*, pp 290–297. Springer-Verlag, 1990.
7. R.A. Brooks. Intelligence without representation. In *Artificial Intelligence*. Elsevier Science Publishers, September 1991.
8. D. Coombs, C. Brown. Real-time binocular smooth pursuit. *Int. Journal of Computer Vision*, 11(2):147–164, October 1993.
9. C. Dunu, J.-Y. Tigli, M.C. Thomas. Software multi-agent system in a graph environment. *CNRS, Lab. I3S*, (96-42), Sept. 1996.
10. V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972.
11. C. Fonseca, P. Flemming. An Overview of Evolutionary Algorithms in Multiobjkective Optimization. In *Evolutionary Computation* , 3(1), pp 1–16, Spring 1995.
12. M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transaction on Robotics and Automation*, 8(3):304–312, June 1992.
13. Y. Gat, J.-P. Müller, M. Rodriguez. Architecture for an Autonomous System : Application to Mobile Robot Navigation. Tech. Rep. 4023-027037, Univ. de Neuchâtel, Suisse, May 1992.
14. B. Horn, B. Shunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
15. H. Mallot, H. Bulthoff, J. Little, S. Bohrer. Inverse perspective mapping simplifies opticla flow computation and obstacle detection. *Biolog. Cybernetics*, 64:177–185, 1991.
16. H.P. Moravec. The Stanford Cart and the CMU Rover. In S.S. Iyengar, A. Elfes, editors, *Autonomous Mobile Robots*, V.2, 407–419. IEEE Comp. Society Press, 1990.
17. S. Negahdaripour, S. Lee. Motion recovery from images sequences using only first order optical flow information. *Intern. Journal of computer Vision*, 9(3):163–184, 1992.
18. F. Panerai, C. Capurro, G. Sandini. Space variant vision for an active camera mount. In *Proc. SPIE AeroSense95*, Florida, USA, April 1995.
19. D.W. Payton. Internalized Plans: A Representation for Action Resources. *Robotics and Autonomous Systems*, 6(1-2):89–103, June 1990.
20. R.A. Regan. Autonomous minehunting and mapping technologies program autonomous maneuvering capabilities. In *Proc. of the OCEAN 1996 Conference*, pp 807–812, 1996.
21. M.J. Rendas, I. Lourtie. Prediction of Positioning Uncertainty Accumulated Over Linear Motion Trajectory Segments Using Dead-Reckoning. *CNRS, Lab. I3S*, (96-43), Sept. 1996.

22. M. Ribo, S. Brahim-Belhouari, L. Pronzato, Motion planning for the precise location of a mobile robot, *CNRS, Lab. I3S*, (96-46), 1996.
23. J. Santos-Victor, G. Sandini. Uncalibrated Obstacle Detection Using the Normal Flow. *Machine Vision Applications*, 9(3):130, 1996.
24. J.T. Schwartz, M. Sharir. A Survey of Motion Planning and Related Geometric Algorithms. *Artificial Intelligence Journal*, 37:157–169, 1988.
25. E.L. Schwartz. Spatial mapping in the primate sensory projection : Analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194, 1977.
26. W. Têtenoire, M.J. Rendas. Planification de Trajectoires d'Exploration pour Robots Mobiles avec des Algorithmes génétiques. *CNRS, Lab. I3S*, (96-45), Sept. 1996.