RESCUE
Cooperative Navigation for Rescue Robots
1st Year Technical Report

A. Bernardino, L. Custódio, P. Lima,
M. I. Ribeiro, J. Santos-Victor, A. Vale

July 26, 2002

1 Introduction

Many researchers around the world are designing mobile robots capable to display increasing autonomy and machine intelligence properties. Most groups concentrate in specific subsystems, such as the planner, the navigator, or the sensor fusion. What usually is missing in their design is a systematic methodology to integrate all these subsystems in a consistent specifications-driven manner, supported on performance (or Quality of Service – QoS) metrics. Extending this to robotic teams would also be desirable and creates new exciting challenges. Such a methodology, should one be available, would help engineering the mobile robots of the future.

Perhaps the key factor of success for a robot lies on its capability to perceive correctly its surrounding environment, and to build environment models adequate for the task the robot is in charge of, from the information provided by its sensors. Different sensors (e.g., vision, laser, sonar, encoders) can provide alternative or complementary information about the same object, or information about different objects, by adequate *sensor fusion* methods. When a team composed of several cooperating robots is concerned, the sensors are spread over the different robots, with the important advantage that the robots can move (thus moving its sensors) to actively improve the cooperative perception of the environment by the team. The information about the environment so obtained can be made available and regularly updated by different means (e.g., memory sharing, message passing) to all the team robots, so as to be used by the other sub-systems.

Another important subsystem is navigation. A mobile robot can move from one place to another by a suitable combination of *local* and *global information* about the surrounding environment, as well as by relying on *metric* (e.g., geometric map, distances to landmarks) or *topological* (e.g., graph representing the spatial relationships among relevant regions) information. When acting as part of a team, navigation can also be improved by cooperation. The location of landmarks of common interest can be improved from their observation by more than one of the teammates. Each team member's absolute location can be made more accurate from its observation by the teammates, which can compute the relative postures. Topological maps can be built by one robot and used by other(s) for navigation purposes. Complementary information from navigation sensors spread over the team robots can also be useful.

Once the information about the world is available and robot navigation is possible, they can be used to make the team behave autonomously and machine-wise intelligently. Three main questions arise for the team:

- Where and which a priori knowledge about the environment, team, tasks and goals, and perceptual information gathered from sensors, should be kept, updated and maintained? This involves the issue of *distributed knowledge representations* adequate to consistently handle different and even opposite views of the world.

- What must be done to achieve a given goal, given the constraints on time, available resources and distinct skills of the team robots? The answer to this should provide a *team plan* and *task allocation*.

- How is the actual implementation of a plan handled, ensuring the consistency of individual and team (sub)-goals, the choice of optimal or quasi-optimal solutions (according to given criteria) and the coordinated execution of the plan? This concerns *task modeling, analysis and coordination.*

The main goal of the "RESCUE – Cooperative Navigation for Rescue Robots" project is to provide integrated solutions for the design of teams of cooperative robots operating in outdoors environments, with special focus in the short and mid-terms on perception and representation issues, as well as cooperative navigation, and, in the mid to long-terms, on task modeling, planning and coordination.

One potential scenario for outdoors robot operation is the search and rescue for victims after a catastrophic event. This may range from natural occurrences, such as earthquakes or floods, to urban riots or terrorist attacks. Many regions of Europe, such as the Balkans, Italy, the South of Spain and Portugal are regions of high or moderate seismic activity.

Earthquakes are specially daunting phenomena. Even though their occurrence is fortunately sparse, the consequences are often tragic, leading to thousands of deaths and many more injured people, besides mass building destruction. Reports from the infamous 1995 Kobe earthquake in Japan show that many unexpected events hit the communications and civil protection infrastructure, disabling the execution of most of the available plans for disaster situations [36]. Especially the serious damage to the telecommunications network and the fact that the buildings of disaster mitigation organizations were hit caused serious delays in the arrival of local and external assistance to the victims. Structures in risk of collapsing are often inaccessible to humans because they are too dangerous. Survivals may be unintentionally injured during removal of debris due to the unawareness of their location or presence by the rescue teams.

Teams of heterogeneous robots can represent an invaluable help for future search and rescue operations. Robots can crawl over the collapsed structures, depositing small-sized robots, and feeding air, water, food and medication to trapped individuals through tubes snaked into the collapsed structure. Small-sized robots can sneak inside very confined spaces, taking with them tiny cameras and other sensors to detect survivors and map survivor locations. Aerial robots can provide a broad view of the search and rescue scenario and map high- destruction locations. Cooperation among human-operated stations, a distributed network of sensors located around the disaster area and teams of tele-operated and/or autonomous robots can increase the amount of available information to the rescue teams. Whenever communications with the networked sensors or the sensors themselves fail due to the earthquake impact, robots can be dispatched to cover the areas most unaccessible to humans and help finding victims at those locations. These mobile robot networks can also provide a dynamic view of the scene at relevant locations.

Currently, the reference scenario for the project refers to a long-term goal of developing robotic teams to help humans in search and rescue missions, and is based on one wheeled and one aerial (blimp) robot. Simplifying assumptions include daylight operation without strong winds.

The scientific advances made during the project lifetime on the enabling disciplines (Computer Vision, Robot Navigation, Control and Artificial Intelligence) should be extendable to other outdoors applications, such as environmental monitoring and surveillance, agriculture or planetary exploration.

This first year report intends to be used as a reference document for the upcoming work to be carried out under the project, and it is organized as follows. Section 2 reports on the work carried out during the project first year, as well as on the changes made with respect to the original proposal. Section 3 summarizes the state of the art of the research on scientific topics close to the project interests, namely representation, navigation, task planning and coordination. In Section 4, the research topics to be developed under the project are detailed. The reference scenario and research roadmaps for the second year of the project are depicted in Section 5.

2 Report on 1st Year Work

The decision of the project reviewers to merge two project proposals (SocRob and Rescue) into one, closer to the original Rescue proposal, forced some changes with respect to the project original research goals and task structure. The most significant result of these changes was the decision to extend the project scope so as to include not only the outdoors (cooperative) navigation and perception research issues, but also the investigation of topics related to task planning and coordination, teamwork and formation control. Even though most of the research has been focused on the original topics, the need arose to view all these topics in an integrated manner.

In the first year of the RESCUE project, thorough scientific discussions took place under the project task T1, as planned, including the plenary presentation of current research issues on Multi-Agent Systems Architectures, Outdoors Navigation, Robot Vision and Task Design and Coordination. The objective of these discussions was three-folded:

- to help the different involved groups sharing a common language;
- to provide an integrated view of the project scientific goals, instead of a simple concatenation of each group contributions;
- to specify a reference scenario that illustrates the research goals, including the required hardware and software.

After these discussions, a scenario, different from the original project proposal, has been approved as the project reference scenario. Its goals are to provide a realistic step towards the long term goal of developing practical rescue robots and to encompass the scientific interests of the project members. The reference scenario, fully described in Section 5, is based on one land and one aerial robot which perform cooperative navigation tasks, some of them requiring formation control.

Another result of task T1 (Design Requirements and Architecture Specification) was a general specification of the software and functional architectures, illustrated in Section 5. The detailed specification is currently undergoing.

The work on task T2 (System Specification, Integration and Assessment) was achieved successfully, including the specification, purchase and integration of the hardware required to accomplish the goals of the reference scenario, notably an iRobot ATRV-Jr land mobile robot and a commercial blimp, to be used as the aerial robot. The latter was purchased using ISR funding, not provided by this project.

The ATRV-Jr is a 4-wheel differential drive robot, with 2 high torque 24V DC servo motors, capable of achieving linear speeds up to 1 m/s and turning speeds up to 120 deg/s. With a 25 Kg payload, it weights 50 Kg and is capable of overcoming 45 deg slopes. Its 2 lead acid, 672 Watt/hr batteries provide an autonomy of 3 to 5 hours, depending on terrain. The onboard computer is a Pentium III running at 800 MHz. Communications with other robots and external computers are ensured by either a 100/10 Mbps Ethernet board or IEEE 802.11b Wireless radio Ethernet. The robot includes an odometry system and Linux-based management software and libraries.

The ATRV-Jr was endowed with several navigation sensors:

- one computerized navigation compass,
- one 12-channel GPS receiver,
- one Inertial Navigation System (INS), including rate-gyros, and
- seventeen sonars (5 forward facing, 10 side facing and 2 rear facing).

The land robot was tested on several runs at IST gardens, with simple navigation algorithms based on odometry and sonars. The GPS, compass and INS were also tested and their performance assessed under several different conditions (e.g., number of available GPS satellites, robot inclination). A report on this work, performed by the company IdMind, is enclosed, detailing the robot hardware and the results of experiments made with it.

The blimp is 4 m long, with a 2 m diameter and 5.5 m^3 volume polyurethane bag. Two of its speed-controlled servo-motors can control altitude, as well as forward and reverse speed, under less than 5 Km/h winds. The remaining servo-motor is attached to the bottom tail and helps controlling the blimp orientation. The payload is 2 Kg. The onboard sensor is a pan and tilt color CCD camera with HF transmiter. Originally, it had a seven channel

41Mhz FM radio transmiter from Graupner (mc-10), now modified to act as a 5-channel PC-Radio Controller transmiter interface using RS-232 communication. The 5 channels are used to control, from a ground computer, the speed, altitude and rotation servos, as well as the pan and tilt system. The ground computer receives the video signal sent through the HF video channel, processes the images and closes the navigation loop by sending commands to the servos and the pan and tilt systems. The modified hardware includes an emergency switch for manual control. The robot is powered by 2 NiMh 7.2v 1900mAh batteries for motors and control, and 8 NiMh 1.2v 600mAh batteries for camera and transmitter power, providing an autonomy of 30–120 minutes.

A report on the blimp, performed by the company IdMind, is also enclosed, detailing the aerial robot hardware, the modifications made and the results of experiments made with it.

Tasks T3 (Navigation) and T4 (Cooperative Navigation) are currently in their initial stage, concerning mainly the design of novel algorithms for vision-based topological navigation and metric navigation based on multi-sensor systems (GPS, electronic compass, gyros, sonars, vision).

A web page has been created for the project, which URL is

http://rescue.isr.ist.utl.pt/.

3 State of the Art

3.1 Task Allocation, Plan Representation and Generation

The advantages of performing a given task by a multi-agent team are partially overshadowed by problems concerning the distribution of tasks between the agents. These are usually hard problems, especially if one looks for an optimal task allocation, due to the number of parameters involved, such as the agents' cognitive and commitment capabilities, as well as their skills, the nature of the tasks, or the social organization of the group of agents. The solution of the task allocation problem can be seen as the answer to the following question [21]: *who has to do what, and using what resources, on the basis of the goals and the skills of the agents and the contextual constraints (type and quantity of resources, nature of environment, etc.)?*. Answers to this question frequently borrow concepts from Artificial Intelligence (due to the social issues involved, such as contract and commitment) and Distributed

Systems (due to its computational counterpart, well-known from problems such as distributing processes by processors).

Ferber [21] classifies the existing approaches to task allocation in two *predefined* modes (*centralised* and *distributed*) and the *emergent allocation mode*. The latter is typical of reactive systems, but we will summarize here only the predefined modes:

centralised allocation mode: requests for services always go through one or more agents. Two sub-modes can be distinguished:

- *hierarchical* — there is a superior agent that orders a subordinate agent to carry out a task. This reminds procedure calls in classic programming and is met only in fixed organisations;
- *egalitarian* — there are special agents (*traders* or *brokers* which manage all the allocation procedures by centralising the requests from clients and the bids for service from servers. Therefore, this centralised mode can be applied to variable organisations;

distributed allocation mode: each agent individually tries to obtain the services it needs. Two sub-modes can also be distinguished:

- *acquaintance networks* — assume that each agent acting as a client has a representation of each other agent's capabilities, by at least having a strongly connected network of contacts;
- *request for bids* — also known as the *contract net*, relies on a 4-stage negotiation process, where i)a manager sends the description of a task to be performed; ii) the bidders, i.e., agents in conditions to carry out the task, submit their proposals to the manager; iii) the manager receives and evaluates proposals and awards the contract to the best bidder; iv) the selected bidder (i.e., the contractor) sends a message to the manager commiting itself to carry out the task or giving up to do so, therefore triggering a re-evaluation of the bids.

The major advantage of the centralised mode is the fact that the trader knows the set of available agents and their skills, facilitating the optimisation problem, as it can more easily select the best agent for a given task request. Drawbacks are the performance bottleneck due to the fact that all decisions go through the trader or manager, and the high sensitivity to failures of the trader.

Contract nets, the most well-known form of distributed allocation, also have their advantages (simple to create, very dynamic, based on bilateral agreements only, which allows increasing the number of parameters taken into account) and drawbacks (generates a large number of messages between agents, management for high-performance allocation may turn out not to be that simple).

Finite state automata (FSA) have been the most widely used plan representation in the literature [22, 37]. In this representation, states represent actions to be performed, while arcs have associated logical conditions over events. A given state has an associated set of enabled events (i.e., those that are monitored under that state). If one or more of these events occur(s) and makes true the logical condition associated with an arc exiting from that state, the plan moves to the action associated with the state the arc points to.

FSA have the advantage of the availability of several tools for analysis and synthesis [35] (which have not been used) but suffer from limited modeling capabilities. Petri nets [35] extend the modeling capabilities of FSA and provide a more convenient modeling methodology starting from the identification of the system components and events. A wide range of analysis (e.g., concerning boundedness, liveness, stochastic and deterministic time) and synthesis (e.g., concerning admissible marked languages) tools is also available, and the non-decidability of some analysis problems can be overcome with no significant expenses. Furthermore, modularity and system design can be achieved by interconnecting several sub-systems, each modeled as a Petri net. This is particularly convenient to model relational behaviors, where more than one teammate is involved. So, Petri nets are being investigated as an alternative tool for behavior modeling. Behavior switching can also be modeled as discrete-event systems supervision [35], for which there are results available regarding FSA and Petri nets. Production systems also have modeling characteristics that make them suitable for this purpose. However, further work must be done to study its design and analysis properties.

Traditional plan generation concentrates on determining how can one agent modify the current world state to another world state closer to the goal state. Even this conceptually simple problem is very difficult to solve, and several simplifying assumptions are usually introduced to make the planning problem tractable [23]: the planning agent knows everything relevant to the planning problem, knows how its available actions can change the world state (e.g., if the world is represented as a set of clauses, as in STRIPS [24], the clauses to be added and deleted after the execution of an action are well-known) and is in control of the world, since only its deliberate actions change the world. Furthermore, the agent's preferred world states are constant throughout a planning episode (i.e., the agent does not change its "mind" about its goals) and the world does not change during the planning process time. This allowed the traditional serialization of the planning problem: first the agent formulates the plan and then executes it. In dynamic and uncertain environments, the weaknesses of this approach are naturally amplified. Alternatives are to enumerate all possible plan contingencies and come up with alternative plans from the very beginning. This is called *conditional planning* or *contingency planning*[23]. If the number of alternatives is too large, *plan monitoring and repair* approaches can be used, where the plan execution is monitored and the plan is modified only when a considerable deviation from the expected route is detected.

However, in complex environments, opportunities for better plans which do not necessarily violate the current plan can happen unexpectedly. The agent's goal(s) can also change continuously. Therefore, the agent should continuously revise its plans, and not wait for triggering events. It should also not plan too much in detail and too far into the future. This modern approach is called *continual planning*, under which plan generation and execution are interleaved.

In *distributed planning*, the planning agent which is part of a team can accomplish its objectives by exploiting the strengths of its teammates through cooperative execution. Even if cooperation is not explicit, one agent can influence other agent's decisions so as to increase the quality of its local plan.

Finally, *distributed continual planning*[25] is an approach to planning where each of the engaged agents is a continual planning agent and the refinement of its decisions over time should take into account the other agents' decisions and refinements.

3.2 Communication Languages and Task Coordination

A consensual issue in multi-agent systems research is the desirability of the agents having interaction and interoperation capacities. To share knowledge, the team agents need mutual understanding and to communicate that knowledge. Therefore, effective interaction and interoperation require three components[26]:

- a common language,
- a common understanding of the knowledge exchanged,
- the ability to consistently exchange whatever is included in the previous items.

Among the existing approaches to the development of communication languages and protocols, the Knowledge Sharing Effort (KSE) consortium is aiming at producing of what may end up being a standard language for agent communications, the Knowledge Query and Manipulation Language (KQML). KQML was conceived as both a message format and a message-handling protocol to support run-time knowledge sharing among agents [27]. It includes communication primitives, special agents called facilitators and the ability to communicate both sentences in some language and an attitude about the content (assertion, request, query).

Agent interaction and interoperation must be coordinated during task execution, similarly to the coordination of the subsystems within each individual agent. However, due to the distributed nature of the agents, as opposed to a more centralized scenario when the subsystems of one agent are concerned, different solutions for task coordination exist in the literature.

A somewhat centralized, but modular, approach is the one based on methodologies for the supervision of Discrete Event Systems (DES) [38]. These consist of observing the events generated by the DES under control (e.g., the robotic task) and disabling controllable events so that the output string of events meets the supervised system specifications (i.e., that the generated strings should be in a language bounded by a minimal required language and a maximum admissible language). The supervisors can be designed modularly, by providing a supervisor for each specification subset and then enabling only the events in the intersection of the sets enabled by the different supervisors.

A decentralized approach can be modeled under this framework when partial observation is assumed for each of the modular supervisors [39, 35]. This way, a supervisor can be designed for each agent, the different observation capabilities of the team agents are considered and the specifications made for the multi-agent system.

3.3 Teamwork

There may be *coordination* among agents without *cooperation*. A good example is the coordinated behavior of drivers in a highway, while there is no specific cooperative goals for them acting as a group. Coordinated behavior is one where the different actions are articulated by rules known of all the team agents. Cooperation arises when teamwork is required, such as in ballroom dancing or playing collective sports. In less obvious situations, it results from the agents' expectation that they will benefit from acting as a group. The theory of joint intentions and commitments [28, 29] addresses teamwork issues, especially how to ensure that, despite the overhead involved in participating in a joint activity (e.g., due to communications requirements), the agent expects to share this load in achieving a common goal in a way robust to failures and misunderstandings among the team agents. For this purpose, logical modeling of individual and joint commitments (defined as a goal that persists over time), as well as individual and joint intentions (defined as a joint commitment to perform a collective action while in a certain shared mental state) is proposed.

Development tools that build up on this and other approaches to cooperation (e.g., Grosz's SharedPlans [30]) have been developed. A popular example is STEAM (Shell for TEAMwork) [31].

3.4 Formation Control

Some multi-robot tasks require the robots to act in a formation. Currently there is a wide body of literature on formation control [40, 43, 41] and related subjects, such as formation feasibility and abstraction [42]. Most methods rely on a follow-the-leader approach, where each member of the formation follows another member and the formation leader follows a pre-specified trajectory [40]. Typical variables to be controlled are the relative distance and orientation between follower and leader. Bounded input bounded output stability, in the sense of ensuring that a stable trajectory of the leader will not lead to unstable trajectories of the follower(s), has been studied [43]. Before controlling a formation, one must make sure the specifications for the formation (e.g., the geometric constraints to relative position and orientation of the formation members) are feasible. This is a particularly intersting problem when the vehicles are non-holonomic. Tabuada *et al* have modeled a formation by a graph where each node represents the vehicle's kinematics and the arcs have associated constraints among vehicles [42]. They provide conditions for formation feasibility under this framework and abstract a formation group when the formation is feasible. The feasible directions of motions for the group are also determined.

3.5 Navigation and Representation

Most of the research on navigation has been centered on the problem of building full or partial 3D representations of the environment, which are then used to drive an autonomous robot. Rather then concentrating the available resources on the actual navigation problem, the emphasis is often put on the process of building and using these 3D maps, requiring large computational power. This is a severe limitation for systems with limited sensory and computational resources.

Studies of animal navigation suggest that most species utilize a very parsimonious combination of perceptual, action and representational strategies that lead to much more efficient solution when compared to those of today's robots. Many animals make alternate use of local/accurate *vs* global/coarse navigation strategies. For instance in [9], there is evidence of animals using alternately landmark-based navigation and (approximate) route integration methods. This *distance/accuracy* trade-off between *long-distance/short-precision* and *short-distance/high-accuracy* mission segments plays an important role in finding efficient solutions to the robot navigation problem. We denote these navigation modes as *Global* and *Local*.

Current global localization methods can be classified as *topological, geometric* or *hybrid*:

- A Topological Map is an adjacency-graph based representation of the environment composed by nodes and links. Nodes represent distinctive places that the system must be able to recognize using external sensors. Edges represent paths between nodes and contain procedural information about going from node to node, using either internal and/or external sensors.

- A Geometric Map is a two/three dimension cell grid, where each cell contains information on its occupancy (deterministic or stochastic). System localization is defined in precise metric terms and it is best suited for use with range sensors (laser, ultra-sound), for data fusion and for accurate motion planning.

- Hybrid Maps are grid based local maps connected by topological relations.

Even though other type of sensors can be used, especially range sensors (e.g., sonar, laser), most modern local navigation methods rely on vision-in-the-loop approaches, where local features, relative optical flow or a portion of a video mosaic are used to guide the robot locally.

In this subsection we discuss how current navigation techniques, both vision-based and multi-sensor based can be used to meet global and local navigation requirements of the RESCUE project. We also refer to Simultaneous Localization and Mapping (SLAM) techniques.

3.5.1 Topological Maps

Topological maps make possible the exploration of unknown and unstructured environments in the absence of metric information (e.g., outdoor environments). The world is represented by a graph (not necessarily 2D) consisting of a finite number of nodes representing regions, linked by bi-directional edges representing paths between regions and/or actions.

A topological map is also a structured feature map (*featured-based map*) which combines the optimal tracking capabilities of feature maps with the scalability of a topological map. Each node is represented by a set of "nearby" features (distinct situations, places or landmarks) and each edge links a pair of nodes. When traversing such a graph, the current node is changed when a the majority of the observed features belong to another node — in what is called nearest neighbor learning.

The advantages of topological maps are scalability, incremental building/update and the need for no map matching. Simultaneously, the uncertainty of environment localization and perception is reduced. Topological approaches do not require the exact determination of the geometric localization of the robot. The resolution is proportional to the complexity of the environment representation. Compactness in a key advantage of this type of representation and low computer processing allows fast planning and facilitates interfacing to symbolic planners and problem-solvers. However, no metric information and feature (or landmark) selection/detection/recognition is possible, which may be a disadvantage. This means that topological representations are heavily dependent on a powerful system to identify key elements of the environment. As a result, one of the most known localization problems using topological representation occurs when the robot traverses two places that look alike. The topological approaches often have difficulty determining if theses places are the same or not, particularly if these places have been reached via different motion commands, actions or paths.

Topological maps are usually considered more resource efficient, because information is represented with accuracy levels adapted to system needs, whereas geometric maps represent detailed metric information everywhere in the map. However, topological maps are ambiguous in the type/amount of information used to represent the nodes and links. Thus there is more freedom for the designer to provide adequate mechanisms to build/navigate on the map.

Most systems to date use geometric maps, mainly because the commonly used sensors in robotics acquire range data (laser, sonar, infra-red). Vision can be an important complementing sensor, motivated by the richness of the visual information and its wide presence in biological systems. In [4] it is shown that many biological systems, ranging from honeybees to humans, use several visual behaviours in their interaction with the environment, but in general visual information is not used to acquire accurate range information. Instead, visual information is used in a more qualitative way with close integration with motion control loops, without explicit use of metric information. Some relevant vision based behaviours found in biological systems are: trail following, aiming at targets, appearance and landmark based guidance, place recognition, among others. For most of these behaviours, environment representations expressed by topological maps are best suited. Topological relations can be expressed qualitatively with relation to explicit sensor data (targets, landmarks, pictorial appearance, tracks, paths, etc) instead of metric information. Furthermore, when considering large complex outdoor environments geometric maps have an exponential growth in complexity, while topological maps are more computationally efficient.

In [6] a topological map is built by driving a robot through an indoors environment and acquiring grey-scale omni-directional images along the path. A principal component analysis (PCA) is performed on the images to compress the data set, retaining only the image components. At place recognition time, the current image is projected on the components of the PCA space and a qualitative localization is obtained by detecting the nearest neighbors. To move from node to node, local navigation methods are used, such as following corridor guidelines and positioning relative to landmarks.

In [1], the image content is enriched by using a colour omni-directional camera. The topological map is acquired manually by driving a robot and grabbing images at a certain rate. These images are analyzed and some representative images are selected to represent the nodes of the graph. The recognition phase is done by comparing images acquired online with the images of neighbor nodes, by histogram matching on individual colour bands of components HLS and normalized RGB. Histograms are compared with Jeffrey's divergence criteria and image is classified by unanimous voting. This method is inspired in some image-database retrieval techniques but is more efficient because comparison is only made with images in a neighborhood of the current location.

Works [5], [6] and [1] show distinct ways of defining distinctive places. While [5] represents places by the geometric relations between environment features (points, edges), [6] and [1] use the global appearance of the scene for comparison with run-time acquired images. Additionally, [6] represents appearance as a PCA basis, which retains geometrical organization of token in the images, whereas in [1] that information is lost by computing the colour histogram of images. The former may have difficulties when dealing with geometric changes in the environment, while the latter may give ambiguous results on environments with uniform colour and luminance characteristics. The combination of both methods in a hierarchical fashion may provide a better solution: the latter would classify the current location in terms of broad classes that present distinct luminance and colour distributions (urban, forest, beach) while the former would perform a more local classification.

Many topological maps are built manually in a training phase, without major concerns on the definition of nodes. Nevertheless, map construction should match the capabilities of the system - the definition of distinctive places should conform with the system ability to recognize and navigate between them. Both Temporal and Spatial distinctiveness are important for selecting "good" nodes for the topological map. Global vision-based approaches do not have to concern with spatial distinctiveness since the whole image information is used. Local (feature) based approaches must select spatially distinct landmarks for node definition and matching. The following referenced papers address issues concerning the problem of distinctive place definition.

In [8], the temporal distinctiveness problem is addressed. Panoramic images are used in a non-structured environment. Distinctive Places (DP) are defined whenever an image is distinct enough from the previously stored images (snapshots). This is computed from the cross correlation between the current image and the snap-shots on the database. A homing behaviour is presented that drives the robot towards a DP provided that the robot is close enough to it (catch area).

In [2], the problem of selecting salient and distinctive features from grey-scale images is addressed. Task independence, domain independence and view point invariance are desirable features for the system. Salient features are selected with the Harris Corner Detector which is robust to small changes in view point. Potential landmarks are characterized by a feature vector containing the brightness and its first and second derivatives at appropriate scale (2-jet). They define functions F of the 2-jet vector that are invariant to view point small changes (differential invariants). The matching function is given by the Mahalanobis distance of F at each template point, where the covariance matrix is computed in a training phase. Recognition is then performed with the Nearest Neighbor criterion. The most distinctive landmarks are the ones that have large Mahalanobis distance to all the others.

In [3], panoramic images are also used but instead of using all image data to describe the places, some landmarks are selected from the image. Landmarks selected in this way can also be used for 'in-place' localization. Landmarks are represented by 16x16 windows. Good landmarks are defined as having good static and dynamic reliability and be distributed uniformly in the image. Static reliability is evaluated by a measure of uniqueness of the landmark in a 32x32 neighborhood, given by the ration of its auto-correlation and the maximum correlation in the neighborhood. Dynamic reliability is measured by the average of the static reliability along a test trajectory involving view point changes. Uniform distribution is enforced by dividing the image in four sections (front, back, left, right) and selecting four landmarks on each section. The best landmarks, according to this criteria are used to represent the place. Matching is performed by a normalized correlation to gain some robustness to illumination changes.

A context-free attentional operator based on symmetry is proposed in [7]. This work is not related to navigation but presents a psychologically motivated operator for focusing attention on the visual field. According to the Gestalt psychology, symmetry is considered one of the basic principles of perception. Our sense of symmetry is so strong that almost every man-made objects are symmetric. Symmetry has been suggested as one of the non-accidental properties of objects which should trigger attention and guide higher level processes. In this sense, symmetry can be used as a property of landmark distinctiveness.

3.5.2 Geometric Maps

The probabilistic approach to the localization problem is typically more robust regarding sensor limitations, sensor noise and environment dynamics as referred in [15]. In [18], Thrun identifies the global localization problem as particularly challenging. Approached probabilistically, the localization problem is a density estimation problem computed using Bayes' rule, the theorem of total probability and the Markov assumption. Within the context of mobile robot localization, the result is often referred to as Markov Localization (ML) or Hidden Markov Models (HMM) [18],[19]. In addition, the Kalman Filter (KF) is the most popular approach to uncertainty handling, representing the belief on the current location by Gaussians, as referred in [19]. The same author developed the Bayesian Landmark Learning (BaLL) algorithm, using Bayes' rule for indoor environment applications [18]). This algorithm enables mobile robots learning what features/landmarks are best suited for the localization process.

In [17], inspired by traditional ship navigation, the authors developed a Coastal Navigation algorithm, using ML as a probabilistic approach. The motivation for coastal navigation is generating trajectories for the robot that reduce the likelihood of localization error. Moreover, proximity sensor such as laser or sonar range-finders have finite range that requires odometry for dead-reckoning. Experimental results were obtained with the Minerva robot in a museum tour-guide robot (indoor environment). These results demonstrated that a probabilistic approach is robust enough for the navigation in dynamic environments such as a museum with a significant people density.

In [14], a partially observable Markov decision process (POMDP) is used to estimate the position of the robot in the form of a probability distribution. The algorithm adjusts the probabilities of the initial Markov model by passively observing the robot interactions with its environment. Learning is unsupervised and passive, meaning that the robot gets no information from a teacher while it is performing other tasks. This unsupervised, passive method is based on the Baum-Welch algorithm, described in [13]. This is a simple expectation-maximization algorithm for learning POMDP from observations.

Most of the work developed using HMM was applied to indoor environments. In [20], Olson describes a probabilistic self-localization technique for mobile robots that is based on the principle of maximum-likelihood estimation. The basic idea is to compare a map generated using any method to detect features (vision, sonar or laser range-finder) at the current robot position with a previously generated map. This method is able to operate in both indoor and outdoor environments using either topological maps or geometric maps (mainly occupancy grids). A drawback is that the method requires a previous map of the environment, as illustrated by the experimental localization results for the Sojourner rover on Mars presented in the paper.

3.5.3 Hybrid Maps

Hybrid maps aim at combining the advantages of topological and geometric maps. Both approaches exhibit strengths and weaknesses. In [16] an approach that integrates topological maps and grid-based maps is described. Since the intrinsic geometry of a grid corresponds to the environment geometry, the position and orientation of the robot in the real world can be determined. Topological maps determine the position of the robot relative to the model based on landmarks or distinct sensor features.

Recent work [5] describes an outdoor vehicle navigating in a park and suburban street environments with a topologically structured feature map. A laser range finder is used to detect features in the environment (points and edges) and topological nodes are defined as spatial arrangements of features. The topological map is selected manually on a training phase. A data association algorithm allows the system to recognize previously trained places and navigation between places is made by metric localization using both odometry and feature tracking data. This somewhat violates the concept of topological navigation, since the representative power of links is not fully exploited. Navigation between links relies on metric information, which can be used to relate all the nodes and obtain a metric map.

The main idea of map building consists of two steps, shown in Fig. 1. The first step is the definition of a topological map as a set of nodes and edges to accomplish the SLAM. Second, each node (as a set of nearby features) has a geometric map, where the robot's pose and the position of each landmark are defined. Each geometric map has a coordinate system that allows the para-

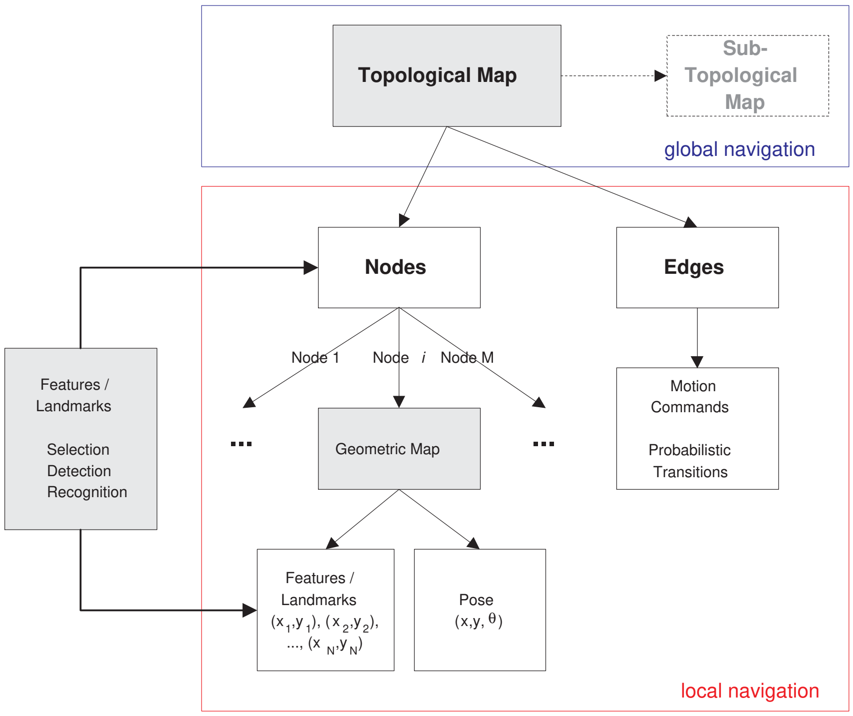

Figure 1: Block diagram of topological map building

metric information to the local navigation, local path planning and obstacle avoidance. Moreover, each navigation planner also requires a specific level of representation as referred in [15]. Both topological and geometric maps require features/landmarks processing, which are entirely sensor dependent. In addition, it is necessary to define the edges between two or more nodes. That connection could be made by motion commands, like go ahead until you find something or move in this direction.

Back to Markov Localization, the underlying probabilistic approach requires the model of the sensors, often non-linear but assuming a Gaussian approximation. In ML each grid cell is identified by a combination of feature values (set of features). For the same reason as explained before, the model is defined by a sum of Gaussians. It still requires a "know how" to identify and update the Gaussian parameters, as referred in [13]. Transitions between cells are defined by motion commands (as previously mentioned) and have associated probabilities. Each cell can be identified as a state, requiring a starting state to initialize the algorithm. The transition probabilities can be related to metric distances between states unless there is no metric information among states. In that case (and without previous information) the probability density function of transitions start as a uniform distribution that is updated through time.

Therefore, the integration of both map representations (hybrid maps) and combining the probabilistic approach to SLAM becomes a challenging problem, mainly in outdoors applications. It still requires a maximum automation, without a priori map, using the minimum of assumptions. The most promising probabilistic approach is ML, that solves the localization problem based on a probability density function. The large number of papers and bibliography that contain theoretical bases is a strong support to use Markov models. In spite of the large number of experimental results documented, most of them belong to indoor results. In addition, the outdoor environments still require a previous knowledge of the environment to determine (or learn) the optimal parameters of the sums of Gaussians. These approximations define the probability density functions that belong to the Markov model. Using Gaussian approximation is an assumption to overcome the non-linear models.

However, the features/landmarks selection problem still remains. From the different types of available sensors, such as sonar, laser, vision, odometry, GPS, compass, inertial, orientation and temperature, and from the observation measurements it is strictly necessary to extract the fundamental information to choose the appropriate features/landmarks and build/update/spread the map. In this way, the topological maps are not dependent of an initial map definition, which is similar to initial human intervention.

3.5.4 Local Navigation Methods

Global navigation methods (e.g., topological) can be used to travel between distant places, without relying on accurate localization along a path. To navigate along a path we must define appropriate navigation strategies, possibly making use of knowledge about the environment geometry. In the RESCUE project we consider both land and aerial robots on a outdoor environment. Therefore we can use some of the available geometrical structure of the environment.In outdoor scenarios, this may include roadway guidelines, external walls, planar surfaces (e.g. the ground floor as seen by an aerial robot at high altitude). Usually, local navigation methods require higher sampling rates and closed loop control to cope with dynamic environments. In this subsection we provide examples of vision-based servoing for robot navigation.

In [10], the centering reflex is described to explain the behavior of honeybees flying within two parallel "walls". The qualitative visual measure used is the difference between the image velocities computed over a lateral portion of the left and right visual fields. The ocular geometry was called the Divergent Stereo, and attempted to use wide field of view images for navigation. The major driving hypothesis is the use of qualitative depth measurements. The goal of the visio-motor controller s limited to the "reflexive" level of a navigation architecture acting at short-range. In spite of these limitations, a variety of navigation capabilities were demonstrated which are not restricted to obstacle avoidance or to the "centering reflex", like the possibility of overcoming unilateral or bilateral absence of flow information, caused by absence of texture or localized changes in environmental structure (e.g.an open door). The system does not critically rely on the accuracy of the optical flow computation, because measurements are continuously used in closed loop. The overall structure of the robot control system involves two main loops. The navigation loop controls the robot rotation heading in order to balance the bilateral flow fields, hence maintaining the robot at similar distances from structures on the right or left sides. The Velocity loop controls the robot forward speed as a function of the amplitude of the lateral flow fields. The robot accelerates in wide spaces and slows down when the environments becomes narrower. Additionally,a sustaining mechanism is implicitly embodied in the control loops to avoid erratic behaviors of the robot, as a consequence of localized (in space and time) absence of flow information.It allows the use of the robot in environments far more complex than corridors, or when the "walls" are not uniformly covered with texture.

In another example, we exploit the parallelism of the corridor/roadway guidelines to control the robot heading direction [6] . To simplify the servoing task, the visual feedback is provided by bird's eye views of the floor, computed from omni-directional images. Tracking the guidelines is done with bird's eye (orthographic) views of the ground plane. We use projective-planar transformations, which can be computed reliably using differential odometric data, to predict the position of points and lines from one image to the next.

Also, in [6] an accurate navigation strategy is explained. When the robot must follow a reference trajectory accurately (e.g. door traversal, docking and navigating in cluttered environments) we rely on *a priori* known templates. Again, bird's eye views computed from omnidirectional images are used to track environmental features, estimate the robot's position/orientation and drive the robot along a pre-specified trajectory. As features, we use corner points defined by the intersection of edge segments, tracked based on a robust fitting procedure. The process of tracking is simplified by utilizing bird's eye (orthographic) views of the ground plane, thus preserving angular measurements and uniformly scaling distances. The robot position and orientation (in the image plane) are estimated by means of an Extended Kalman filter.

For aerial or underwater robots driving at altitude, planar surfaces are good approximations for the geometry of the ground/sea floor. We have developed fast algorithms for tracking planar patches [12], which can be used for servoing in local planar surfaces. In [11] we give examples of using video mosaics as a maps for the navigation of underwater or aerial vehicles. The vehicle pose is estimated with respect to the mosaic, thus allowing us to control its trajectory towards a desired configuration.

4 Research Topics Under the Project

The following subsections cover different research topics which will be investigated under the project by the intervening groups.

4.1 Task Planning and Coordination

Most existing Robotics applications rely on tasks designed "by hand". However, a more appropriate approach would be to develop a planner capable of periodically (or when invoked) analyzing the world state and providing a new set of individual and relational (i.e., involving more than one robot) behaviors that lead the robot (or the team) closer to the goal state. Research under the project is concerned with the following related problems:

Distributed Continual Planning: methods for continuously interleaving plan generation and coordinated execution, distributed by the team members. One possible way of designing plans which continuously adapt to new situations by tuning to the actual surrounding environment is to use reinforcement learning (RL) algorithms, especially those which guarantee convergence properties, such as Q-learning. However, learning is usually slow. An envisaged approach that overcomes this problem is to provide plans with alternative paths among which the RL algorithms can learn to switch over time. Cooperative learning arises when a robot takes its decisions from information learned and provided to it by its teammates. This shared information may range from the current policy to the current view of the world or both. These issues will also be investigated under the project.

Task Breakdown and Allocation: (possibly logic-based) methods to determine the set of primitive tasks that span a given task space and to distribute tasks among the team robots, e.g., providing "satisficing" (in Nash's sense) solutions for the problem of achieving a goal, given the time and resource constraints and distinct skills of the team robots.

Distributed World Modeling: where and which prior knowledge about the environment, team, tasks and goals should be represented, kept, and maintained, how should symbolic knowledge and non-symbolic information be linked together.

4.2 Task Modeling, Analysis and Design

We claim that a consistent model for robotic tasks is required to provide a systematic methodology for behavior synthesis and analysis. Modeling robotic tasks as discrete event and/or hybrid systems by using (hybrid) automata or Petri nets methodologies and analysis tools provides such a methodology that enables formal verification and quantitative performance evaluation. Furthermore, it opens opportunities for novel research on task design from specifications (e.g., reliability, cost, utility), tools for coordinated task execution, and interfaces for task design, due to the inherent modularity of the referred approaches, which allow interconnections among models of individual robot subsystems, as well as among models of individual robots. Under the RESCUE project, the modeling of robotic tasks using discrete-event systems will be investigated.

4.3 Formation Control

Some (not necessarily all) multi-robot cooperative tasks require the robots to act as a formation. We will concentrate on follow-the-leader methods for a reduced number of robots (2-4), with special attention to approaches that handle:

- non-holonomic robots with 6-dimensional configuration space keeping formation stability;
- obstacle avoidance by a sub-group of the formation without breaking the formation.

4.4 Map Representation and Navigation

Since our application considers large complex outdoor environments, we propose the use of topological maps for robot navigation. The following is a list of desirable system capabilities, to be developed under the project:

Map Creation: The system must have the capabilities to create a topological map of the environment. This may be done manually or automatically, at initialization phase or incrementally as the agent progresses in the environment. These capabilities include methods to detect distinctive places (create nodes) and represent the procedural information to go from node to node.

Localization: The system must have the capabilities to recognize/associate distinctive places from the information contained in the map and keep a qualitative localization along the edge. Ideally the system should be able to deal with uncertainty, creating likelihood estimates of its localization within the map and keeping several likely hypotheses in parallel to choose between alternatives.

Guidance: The system must have the capabilities to execute the procedural information contained in the graph edges. Also should have the basic guidance capabilities for safety (e.g., obstacle avoidance and velocity control).

Path Planning: The system should be able to decide between alternatives e.g. when there is more than one edge leaving a node or different paths leading to the same destination. This problem becomes more complex when several agents are involved, as it raises Task Allocation issues (see above).

Communication: Whenever multiple cooperating robots are involved in accomplishing a task, they need to share their absolute postures, as far as navigation is concerned. This may require the need to develop a common spatial representation of the environment. An alternative is to use relative postures. In both cases, communication protocols must be developed (or imported from work on communication networks) so as to ensure team QoS.

4.5 Cooperative Navigation

When a robot team is performing a task in collaboration, the navigation of each of the teammates can profit from the distributed sensing capabilities and different locomotion and positioning capabilities. Localization and motion strategies are problemas that can benefit from a collaborative interaction among robots.

Research will be carried out in the following directions:

- Cooperative Localization using probabilitisc approaches, under sensor and and environment map uncertainties and diversity in sensor working principle,

- Analysis of trade-offs between efficiency and accuracy when using co-operative localization schemes in a team of robots,
- Multi-robot collaboration for robust navigation, in what concerns path planning and path following to reach a giving taget location.
- Multi-robot collaboration for environment mapping and exploration.

4.6 Vision-Based Navigation

Topological representation of the environment may be the most suited environment representation for use in outdoor environments. Edges between nodes represent procedural information (actions) to travel along it. Nodes represent distinctive places (DP) in the environment, e.g., salient landmarks, path bifurcation. Both the detection and recognition of DP's are currently active research topics. A non-exhaustive list includes issues such as what places are distinctive, how to compress information without degrading the recognition capability, how to upgrade the representation in the presence of new information.

Concerning landmark selection, good landmarks can be used for local navigation tasks or to define distinct navigation tasks and focus on relevant parts of the visual field. Salient landmarks are easily identified (pop-out) and distinctive landmarks are unlikely to be confused.

Panoramic (omni-directional) cameras provide a rotational invariance of the field of view. The large field of view reduces the number of images needed to represent a region, making the system more robust to small changes in the environment, like occlusions, object removal or insertions. However, the resolution of such images becomes worse than regular images because more visual field must be represented by the same amount of pixels.

Local servoing strategies are important components of robot navigation systems. Most work done to date considers indoor settings. For outdoor applications, issues like robustness to illumination changes, lack of structure in the environment and unexpected events require special care. Robustness can be increased by using information coded with redundant cues, such as texture, colour, orientation or curvature. Each feature may be more reliable to particular observation noise than others.

Research will be carried out in the following directions:

- Use of visual perception for constructing topological maps and environment representations.

- Analysis of visual geometries (sensor layout) in the context of visual perception tasks, i.e. matching the functional objectives and sensor design.
- Visual servoing strategies for the control of flying and land vehicles.
- Control of multiple heteregeneous vehicles based on their mutual visual perception.

5 Reference Scenario and Roadmap for the 2nd Year

The reference scenario consists of two robots: one aerial blimp/zeppelin robot; one land outdoors robot. The aerial robot will perform several tasks, such as making a visual topological map of the destroyed site. The map will include information on the relevance of each of the mapped locations concerning degree of destruction, presence of victims, etc, as well as on the difficulty of traversing regions between them, due to the presence of debris or obstructed paths. The map will be stored as a graph and will be used to choose the best path for the land robot to reach a goal location (e.g., one with a larger number of victoms). It can also be used to help the aerial robot navigating. The land robot will use several sensors (GPS, inertial, vision, sonars) to navigate towards the goal, handling the details associated to the path (e.g., debris, trees, people on the way, etc). While the land robot moves towards the goal location, the aerial robot should follow it using a formation control algorithm, so as to keep a reliable communications link and to serve as a relay for informations that the land robot may need to send to distant stations.

An animation illustrating the scenario above was developed to better explain the reference scenario to project members and non-members. An mpeg with it can be downloaded from the web site

http://rescue.isr.ist.utl.pt/videos/rescue.avi.

Considering the long term goals of the project, this and future scenario extensions (e.g., by adding more robots) require a *software architecture* capable of handling real-time distributed control and supervision systems, as well as a *functional architecture* to integrate all the required subsystems (and their corresponding functions) consistently.

Figure 2 depicts the diagram of the proposed software architecture to be implemented by the grantee of the project in its remaining two years. It is based on a multi-thread and distributed blackboard approach. The different threads handle sensor data acquisition and processing, communications, behavior execution and behavior switching. Raw and processed data go to the distributed blackboard, a distributed shared memory organized in classes of variables corresponding to the different sensor classes. Some variable values are kept locally at the robot which acquired and processed the data, others are broadcasted (or sent individually) to the teammates. Some transducers have an associated set of virtual sensors, called *modes*. A typical example is the vision transducer, with modes such as locate object, determine object speed, classify object.

An overall view of the functional architecture is shown in Fig. 3. It comprehends the following subsystems: task planning (to handle task/behavior allocation, intra-team communications and formation topology, when a formation is required), task coordination (to handle the decisions on which behaviors to select at every time step, so as to optimize the performance in terms of time taken, reliability, cost, utility or other factors - requiring performance feedback from each subsystem) and behaviors (individual and relational - those involving relations among more than one teammate). A more detailed view of the behaviors subsystem is displayed in Fig. 4. It shows that behaviors can be built as finite state automata, where arcs are associated to conditions over detected events (not shown) and most states correspond to dynamic control systems implementing navigation functions, parameterized according to the task at hand (e.g., move until a given posture is reached, tracking an object or a teammate). The remaining states do not correspond to navigation functions (e.g., Turn Light On) but can also be designed as closed loop controllers (e.g., keep turning on the lights until a light sensor confirms they are on) for consistency and robustness.

Note that the behaviors can be implemented some other way (e.g., based on Petri nets or production rules), with relative advantages concerning the representational power but a potentially reduced analysis power. This will be an issue to be investigated under the project.

The navigation functions are the key practical developments expected during this project. They will be based on a multi-sensor system (GPS, compass, odometry, sonars, vision and inertial navigation system composed of rate-gyros and accelerometers) for the ATRV-Jr, mixing metric and topological navigation. The navigation of the blimp will be mostly vision-based and non-metric, using topological maps.

The current work concerning metric navigation under the project builds on existing closed loop solutions which:

- use GPS as a reference and odometry plus accelerometers as feedback sensors for **position** estimates (x, y, z). The GPS is not as accurate as the feedback sensors but is not subject to drift and can be sampled at a slower rate to reset the position information.

- use the compass as a reference and the rate-gyros as feedback sensors for **orientation** estimates (roll, pitch and yaw). The compass is not as accurate as the feedback sensors but is not subject to drift and can be sampled at a slower rate to reset the orientation information.

The novelty here is on handling particular problems of outdoor environments, such as very poor odometry and frequent loss of GPS information due to trees or buildings. The quality of metric navigation can be improved by cooperative navigation where a network of communicating sensors assembled on the team robots can take advantage of their space diversity to provide information to each of the team robots which they might miss if operating alone (e.g., one of the robots does not have GPS available but one of its teammates does and can also estimate their relative postures). All this work will necessarily require sensor models, which can not be obtained before outdoor runs, during which data from all navigation sensors is registered, are performed. The recorded data can subsequently be subject to statistical analysis in order to build the sensor models.

The current work concerning topological navigation under the project builds on existing approaches to traverse a region by identifying successive images representing a mosaic of sub-regions previously obtained.

The research work will concentrate on low complexity representations of the mosaic images which can still allow the determination of where the vehicle is with a given accuracy (typically lower than for metric navigation). Its novelty comes also from the fact that the mosaic is constructed by the navigating robot itself or by one of its teammates (e.g., the blimp builds a topological map for the ATRV-Jr navigation). The sharing of topological maps among teammates with very different postures and world views will also raise challenging representation research issues.

This project approach is based on an integrated view of the overall system to be developed in the short and long terms. As such, a relevant issue concerns sensor data fusion and the availability of (raw or processed, including fused) data in a distributed blackboard, as represented in Fig. 5. Information on features relevant for the system is obtained in general from different sensors or transducer modes, by fusing them according to sensor models (e.g., considering their measurement uncertainty) and stored in the blackboard. The role of the blackboard is two-folded, as already pointed out when describing the software architecture:

- to provide a sensor database where all the relevant raw and processed data (e.g., from a given time period) is organized and available to all decision levels;
- to share information among teammates, including data not acquired and/or processed by the team member using it.

This distinguishes our approach from non-integrated ones, where data is acquired directly by each subsystem from the sensors relevant for that subsystem, disregarding the fact that this information may be relevant for other subsystems. A typical example is the information on the presence of one or more obstacles near a robot. This information is immediately required for the robot to avoid the obstacle(s) but it is also relevant to update the team world model, needed for decisions at the task planning subsystem.

A software tool to develop multi-robot systems based on the above described software and functional architectures is also currently under design and will be implemented by the project grantee. It is expected that the resulting tool will dramatically accelerate the implementation and test of such systems, not necessarily for search and rescue applications only.

The following list summarizes the implementation work to be made, in the format of a roadmap for the remaining two years:

1. control loops for ATRV-Jr navigation (includes characterization of the navigation sensors);
2. cooperative navigation algorithms, possibly between the ATRV-Jr and some other available robot(s), for the sake of concept demonstration;
3. topological map generation by the blimp and blimp topological navigation;
4. make ATRV-Jr follow the topological map generated by the blimp (includes conversion of map representations);
5. formation control algorithms for the blimp plus the ATRV-Jr;
6. tool for design, development and test of behavior-based multi-robot tasks.

References

[1] Iwan Ulrich and Illah Nourbakhsk, "Appearance-Based Place Recognition for Topological Localization" *ICRA 2000*

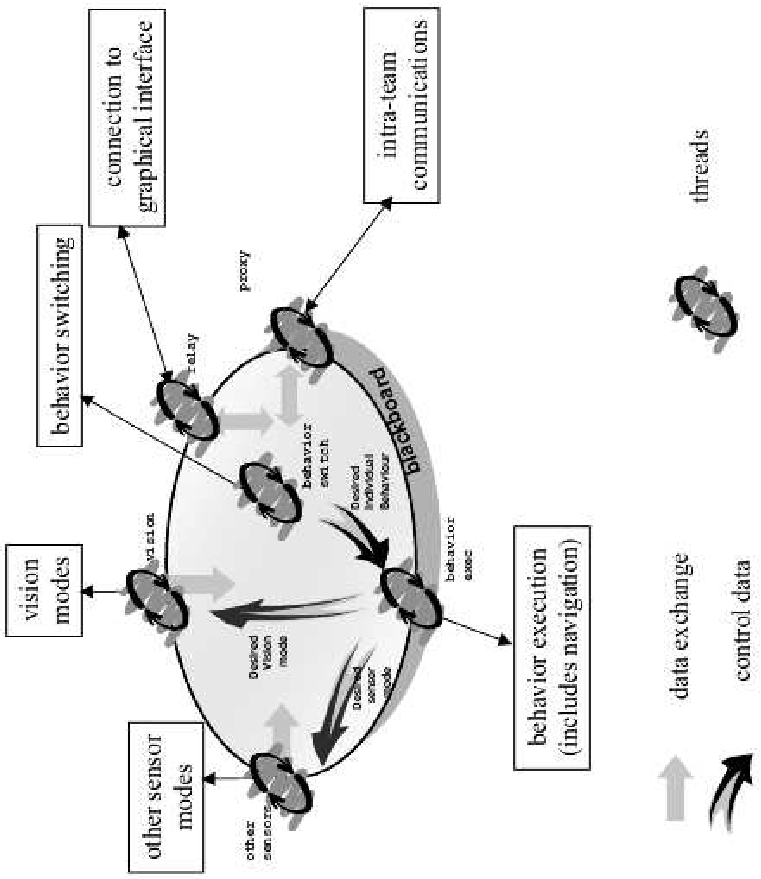

Figure 2: Reference software architecture.

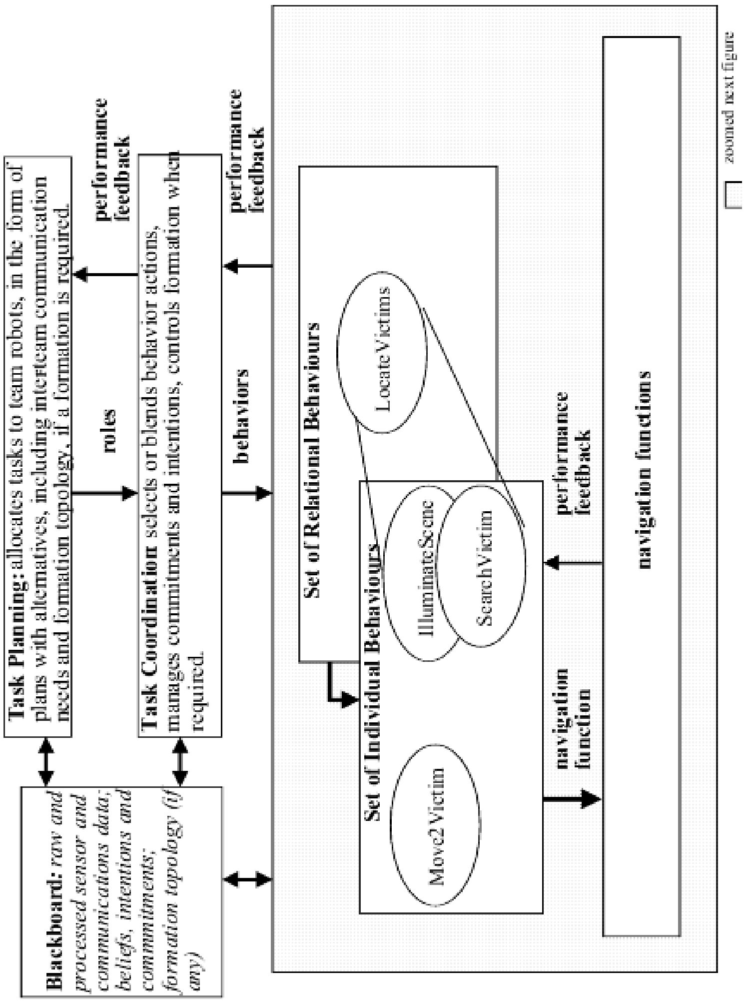

Figure 3: Functional architecture: global view.

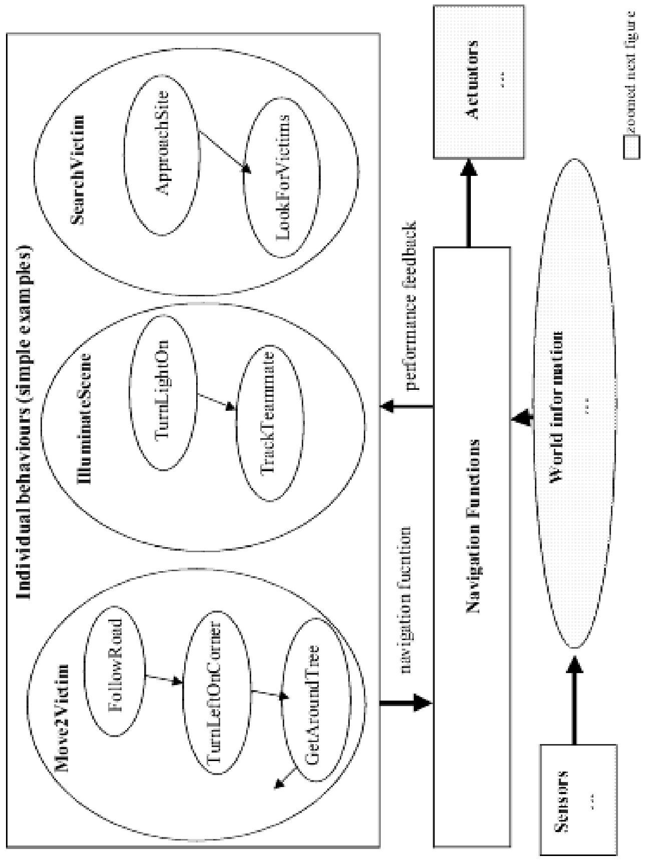

Figure 4: Functional architecture: the behaviors.

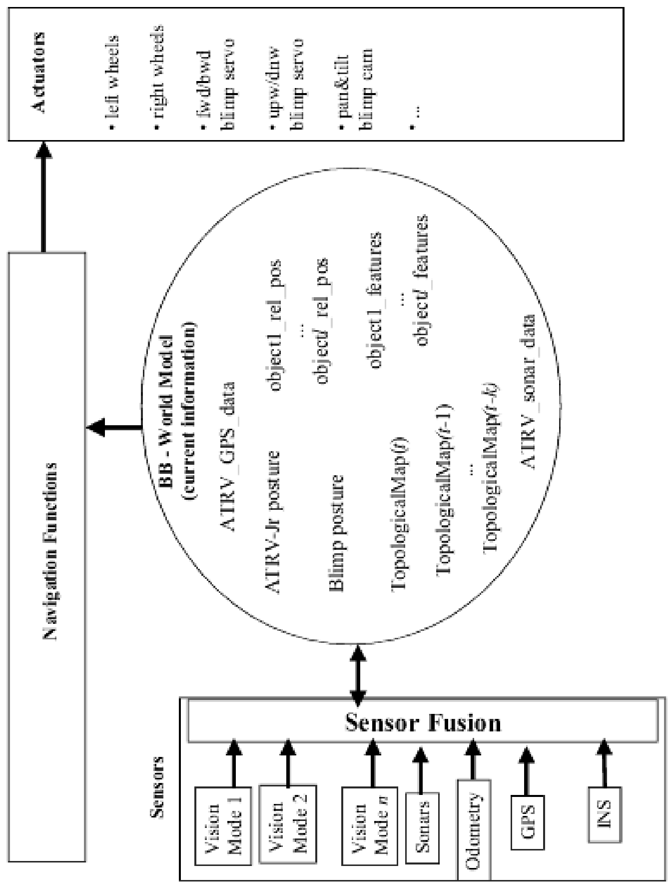

Figure 5: Sensor fusion, world model and the blackboard.

[2] Markus Knapek, Ricardo Oropeza, David Kriegman, “Selecting Promising Landmarks” *ICRA 2000*

[3] Simon Thompson, Toshihiro Matsui, Alexander Zelinsky, “Localization using Automatically Selected Landmarks from Panoramic Images” *ACRA 2000*

[4] M. Franz, H. Mallot, “Biomimetic Robot Navigation” *RAS 30 (2000)*

[5] T. Bailey, E. Nebot, “Localization in large scale environments” *RAS 37 (2001)*

[6] J. Gaspar, N. Winters, J. Santos-Victor, “Vision Based Navigation and Environmental Representations with an Omni-Directional Camera” *IEEE TRA 16 (6), Dec. 2000*

[7] D. Reisfeld, H. Yeshurun, “Context Free Attentional Operators: the Generalized Symmetry Transform” *Int. J. of Computer Vision, Special Issue on Qualitative Vision, 1994*

[8] M. Franz, B. Scholkopf, H. Mallot, H. Bulthoff, “Learning View Graphs for Robot Navigation” *Autonomous Robots 5, 1998*

[9] R.Wehner and S.Wehner, “Insect navigation:use of maps or ariadnes thread?” *Ethology, Ecology, Evolution, vol.2, pp.2748, 1990*

[10] J.Santos-Victor, G.Sandini, F.Curotto, S.Garibaldi, “Divergent stereo in autonomous navigation: From bees to robots” *Int.J.Computer Vision ,vol.14,no.2,pp.159177,1995*

[11] N. Gracias, J. Santos-Victor, “Underwater Mosaicing and Trajectory Reconstruction using Global Alignment” *IEEE OCEANS 2001, Honolulu, November 2001*

[12] Sjoerd van der Zwaan, A. Bernardino, J. Santos-Victor, ”Vision based station keeping and docking for floating vehicles” *ECC2001, Porto, Portugal, September 2001*

[13] L. Rabiner ”A Tutorial on Hidden Markov Models” *Proceedings of the IEEE, vol. 7 no. 2, 1989*

[14] S. Koening, R. Simmons ”Unsupervised Learning of Probabilistic Models for Robot Navigation” *In Proceedings of the International Conference on Robotics and Automation, pages 2301-2308, 1996*

[15] R. Chatila, S. Lacroix, M. Devy "Autonomous Outdoor Mobile Robot Navigation" *RAS 1997*

[16] S. Thrun, A. Bucken "Integration Grid-Based and Topological Maps for Mobile Robot Navigation" *Proceedings of the 13th National Conference on Artificial Intelligence AAAI, Portland, Oregon, August 1999*

[17] N. Roy, W. Burgard, D Fox S. Thrun "Coastal Navigation - Mobile Robot Navigation with Uncertainty in Dynamic Environments" *In Proceedings of Conference on Neural Information Processing Systems (NIPS), 1999*

[18] S. Thrun "Bayesian Landmark Learning for Mobile Robot Localization" *Machine Learning Machine Learning 33(1)*

[19] S. Thrun "Probabilistic Algorithm in Robotics" *AI Magazine, 21(4):93–109, 2000*

[20] C. Olson "Probabilistic Self-Localization for Mobile Robots" *IEEE Transactions on Robotics and Automaton, vol.16 no. 1, February 2000*

[21] J. Ferber "Multi-Agent Systems" *Addison-Wesley, Reading, MA, 1999*

[22] J. Albus "Outline for a Theory of Intelligence" *IEEE Transactions on Systems, Man and Cybernetics, vol. 21, no. 3, May/June 1991*

[23] S. Russell, P. Norvig "Artificial Intelligence - A Modern Approach" *Prentice Hall, 1995*

[24] "R. Fikes, N. Nilsson "STRIPS: a New Approach to the Application of Theorem Proving to Problem Solving *Artificial Intelligence Journal, 2(3-4):189-208, 1971*

[25] M. desJardins, E. Durfee, C. Ortiz, Jr., M. Wolverton "A Survey of Research in Distributed, Continual Planning" *AI Magazine 20(4):13-22, Winter 1999*

[26] T. Finin, Y. Labrou, J. Mayfield "KQML as an Agent Communication Language"

[27] T. Finin et. al. "Specification of the KQML Agent Communication Language" *DARPA Knowledge Sharing Iniciative External Interfaces Working Group, 1993*

[28] P. Cohen, H. Levesque "Teamwork" *Technical Report 504, Computer Science Dept, Carnegie Mellon University, March 1991*

[29] P. Cohen, H. Levesque "Intention is Choice with Commitment" *Artificial Intelligence Journal, 42:213-261, 1990*

[30] B. Grosz, S. Kraus "Collaborative Plans for Complex Group Action" *Artificial Intelligence Journal, 86:269-357, 1996*

[31] M. Tambe "Teamwork in Real-World Dynamic Environments" *Tech. Rep. ISI/RS-96-446, USC/Information Sciences Institute, May 1996*

[32] Stergios I. Roumeliotis, George Bekey "Collective Localization: A Distributed Kalman Filter Approach to Localization of Groups of Mobile Robots", *Proceedings of ICRA 2000, San Francisco, CA, USA, pp. 2958-2965*

[33] R. Madhavan, K. Fregene, L. Parker "Distributed Heterogeneous Outdoor Multi Robot Localization" *Proceedings of ICRA 2002, Washington, USA*

[34] S. Thrun "A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots" *International Journal of Robotics Research, 20(5), 335-363, 2001*

[35] S. Lafortune, C. G. Cassandras "Introduction to Discrete Event Systems" *Kluwer Academic Publ., 1999*

[36] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, S. Shimada "RoboCup Rescue: Search and Rescue in Large-Scale Disasters as a Domain for Autonomous Agents Research" *Proceedings of IEEE International Conference on Man, Systems and Cybernetics, 1999*

[37] P. Lima, L. Custódio "Artificial Intelligence and Systems Theory Applied to Cooperative Robots: the SocRob Project" *Proceedings of Encontro Científico do Robótica 2002 - Festival Nacional de Robótica*

[38] P. J. Ramadge, W. M. Wonham "The Control of Discrete Event Systems" *The Proceedings of the IEEE, Vol. 77, No. 1, pp. 81–98, January 1989*

[39] K. Rudie, W. M. Wonham "Think Globally, Act Locally: Decentralized Supervisory Control" *IEEE Transactions on Automatic Control, Vol. 37, No. 11, pp. 1692–1708, November 1992*

[40] R. Fierro, P. Song, A. Das, V. Kumar "Cooperative control of robot formations" *submitted to the Kluwer Series on Applied Optimization, March 2001*

[41] R. W. Beard, J. Lawton, F. Y. Hadaegh "A Coordination Architecture for Spacecraft Formation Control" *IEEE Transactions on Control Systems Technology, Vol 9, No 6, November 2001, p. 777-790*

[42] P. Tabuada, G. J. Pappas P. Lima "Feasible Formations of Multi-Agent Systems" *Proceedings of the American Control Conference, Arlington, VA, 2001*

[43] H. G. Tanner, G. J. Pappas "Formation Input-to-State Stability" *15th IFAC World Congress on Automatic Control, Barcelona, Spain,July 2002*