Intelligent Systems for Search and Rescue

Shahram Bahadori, Daniele Calisi, Andrea Censi, Alessandro Farinelli, Giorgio Grisetti, Luca Iocchi, Daniele Nardi
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
E-mail: `<lastname>@dis.uniroma1.it`

***Abstract*— Search and Rescue scenarios offer a wide variety of issues for analysis and experimentation in robotics, from mechanical architecture up to AI research themes. In this paper after briefly describing our robot configuration, we address the main issues of autonomous rescue tasks. Moreover, we explain algorithms and methods used by our modules in order to achieve a semi-autonomous exploration and victim detection system.**

***Index Terms*— Search and Rescue Robots, Intelligent Systems.**

I. Introduction

Using robotic technology in search and rescue missions for assisting the rescuers covers a large part of the robotic and artificial intelligence research themes. This type of applications involves different research areas, from mechanical design and sensors interpretation to perception analysis, decision making, mapping, path-planning and victim detection.

The unstructured and unstable environment of the rescue scenario makes the mechanical and locomotive design of the robot critical. The reliability of the communication between the robot and the rescue operator represents a discriminating factor on which autonomy level to choose. The worse the communication, the higher the required autonomy level of the system. Considering that in rescue scenarios the communication can be extremely unreliable, the weight of the autonomy for evaluating the performances of a rescue system becomes high.

Some basic activities that the rescue robot should perform autonomously are: mapping, navigation, victim detection and victim localization. These tasks give the robot the possibility to continue its mission alone and also let the rescuers have more detailed information from the robot.

Moreover, a fundamental reason for increasing the autonomy of a rescue robot is to help the operator in interpreting the information coming from the on-robot sensing devices. Rescue robots are usually equipped with an high number of sensors, and it can be difficult or impossible for an operator to deal with the high amount of data produced by the devices. It should be more useful to provide the operator with an high level description of the environment. Moreover, a team of semi-autonomous robots could be driven, or just supervised, by a single operator.

Hence our goal is to provide a high level interface to an operator, who can only play the role of a planner (from a goal list) or a navigator (taking robot control when needed), in both cases performing his job using easy human readable data, like a map.

In this paper we first describe the rescue robot configuration and the experimental environment on which we have tested our methods. Then we address the software architecture we have implemented. Finally we provide detailed descriptions of the methods used for navigating and exploring the environment, building accurate and consistent maps and detecting the victims in a semi authonomous way.

II. Robot configuration and experimental environment

Experiments have been performed with a Pioneer ATX robot (see Figure 1) in a Rescue arena built at the ISA laboratory in Rome and the implemented system was used during the RoboCup 2004 Real Rescue competition. This robot has four driving wheels and it is able to move over small obstacles. Moreover it is equipped with both a laser range finder and a stereo camera and it is thus suitable for experimenting localization and mapping techniques in both planar and non-planar environments.

The Robot is equipped with a SICK laser range finder, frontal rear sonar rings and stereo vision system, in addition for human body detection we have an Infra Red non-touch thermo-sensor and the voice transmission system. For on board computation we use a Pentium M Laptop and 802.11a wireless communication.

III. System functionalities

We main tasks of a rescue robot are: 1) **exploration and map building**, 2) **victim detection**.

First, the robot needs to plan a sequence of moves and to execute them ensuring successful coverage of the area to explore. Exploration relies on some basic functionalities:

- *Mapping*, provides the map of the explored environment.
- *Localization*, provides the pose of the robot in the map,
- *Path-Planning*, computes a path between two points in the map,
- *Control*, allows the robot to follow a path.

Navigation module has to deal with problems like dynamic obstacles, map incompleteness, movements that unexpectedly fail, and so on, and supply the functionality of moving the robot to goal positions and reports possible failures.

Fig. 1. Rescue robot and arena used for experiments

Furthermore, the localization and mapping tasks are typically coupled in a unique task, that performs simultaneously the robot localization and the map building (SLAM). The SLAM task has an internal representation, which is suited for the requirements of the algorithm, but could not fit navigation and/or operators needs. Therefore, besides the SLAM module, we have implemented a mapping module that integrates the sequence of estimated position supplied by SLAM with all sensor readings (i.e. laser scanners, stereo cameras, sonar) in a typical occupancy grid to provide a map to both navigation modules and operators.

Second, one of the main challenges for robots involved in rescue missions is to identify victims and report their location in the explored environment, for which a map is essential. The victim detection module is based on the use of a stereoscopic camera providing 3D information about the scene and on other sensors integrated in the system to measure the temprature and hear the voice of the victims.

In order to integrate effectively all these modules, we have used a modular software architecture [9], that allows for an effective and efficient integration of different modules implementing the basic capabilities of a mobile robot and for easy reusing and team development. The modular architecture also allows for easily interchange modules as well as connected devices, ranging from actual robots to simulators. Modules are loosely connected to each other and can be scheduled independently with different priorities; interaction and communication among them occur using a centralized blackboard-type data repository. The shared data can be exported to a remote console and to team mates via a TCP link. The operator can visualize data (camera, sensor readings) and edit them (map correction). All modules publish their parameters and internal state which are useful both for debugging and supervision.

IV. Mapping and localization

SLAM has been deeply investigated by the robotic community. It is considered a complex problem, because for localization a robot needs a consistent map and to acquire the map it needs a good estimate of its location. This mutual dependency among the pose and the map estimates makes the SLAM problem hard, requiring to search a solution in an high-dimensional space. Although several effective techniques have been proposed so far, a general one that works with any sensor setting and in arbitrarily unstructured environments is still under investigation.

Rescue environments present some additional difficulties that are not considered in most of the SLAM approaches. First of all these environments are supposed to be less structured than the traditional ones, and this introduces additional difficulties, since the a priori knowledge of the environment cannot be exploited by the SLAM process. Second, most of the SLAM approaches have been conceived for operating in planar working space, while the third dimension cannot be ignored in a rescue environment. Third, building a map of a rescue environment requires to integrate the output of different sensors that can be used in the mapping process, since it is not possible to use a single sensor for detecting all of the materials in a rescue arena, and accurate enough for on line building of maps. For instance the laser range finder, that has been successfully used for building metric maps of office-like environments, fails in detecting glass and mirrors, while the sonar, even if significantly less accurate, can detect these materials. Finally, most of the SLAM approaches focus on having maps that are accurate enough for being interpreted by a program, rather than by a human. To this end a crucial problem is to provide consistent maps, i.e. maps in which sensor informations in different time frames are not conflicting. This is particularly evident when a robot revisits a known place after navigating for long in unknown terrain. The error accumulated in the unknown area is usually big enough to introduce an unrecoverable error that does not allow to correctly detect previously seen location thus generating an inconsistent topology.

Even if most of the works on SLAM deals with the loop closure problem (that in fact represents a fundamental issue, since only when revisiting known locations the error accumulated when mapping new areas can be reduced), the typical rescue arena settings does not require to deal with such a problem (especially in planar environments), due to the reduced size of the arenas and to the fact that, exploiting the precision of accurate sensors like the laser range finders, the error can be bounded so that no signifi-

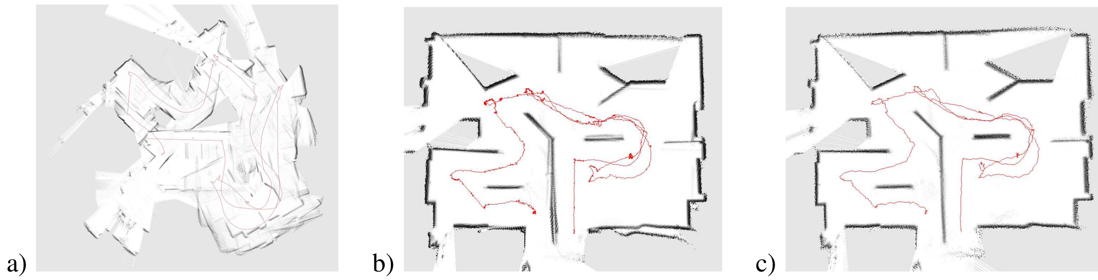

Fig. 2. a) Occupancy grid generated by odometry data data taken in the yellow rescue arena. The odometry is extremely bad. b) ScanMatch result. c) The Rao-Blackwellized mapper results. In this case the map produced by Rao-Blackwellized algorithm is more consistent.

cant inconsistencies arises. Moreover, when interpreted by human operators, small errors in loop closures typically do not cause any misinterpretation of the map.

A. *Mapping method for planar environments*

As many works on SLAM have shown [14], [15], [27], [21], [20], the best setting for building maps in indoor environments is to use a wheeled robot equipped with a laser range finder. Even if such a sensor is not able to detect some kind of surfaces (like mirrors and glasses), its high accuracy allows to build accurate planar maps. However, laser range finder cannot deal with any material; for example glasses and mirrors (that are typically present in a rescue environment) introduce large errors in the mapping process when only a LRF is used. In order to deal with this aspect we have used also ultrasonic sensors, that, although less reliable and accurate of LRF, allow for dealing with these materials. The strategy we have adopted has been to discard all of the laser readings falling in the sensor cone, whose values are greater than the corresponding sonar reading. In this way we are able to recover several situations in which the LRF cannot detect a glass or mirror element in the environment.

From the analysis of the common techniques for SLAM, we have devised and implemented a mapping method that is suitable in rescue environments. The method is an incremental scan matching technique that aligns each scan with the previously accumulated ones, by performing a local search around the odometry estimated position, trying to maximize the past and current scan overlapping. As a difference from the Lu and Miles approach our scanmatcher does not rely on the correspondence among the single readings. Our approach is based on the maximization of a function of the robot pose, given the current sensor reading. The more the scan overlaps with the map built so far the higher the value of the function. It is very similar to the approach used by the carmen scanmatcher (vasco) [1], but presents some differences in the map representation: in our implementation the map is fixed, while in vasco it is computed at each frame time step, based on the readings history. This results in an increased speed since the cost for building the reference map is constant instead that linear in the number of accumulated readings as in vasco.

B. *Results*

The experiment have been performed by comparing our method based on scan matching and Rao-Blackwellized mapper [13], [29]. The results show that Rao-Blackwellized is more accurate when closing loops, while our scan matching method is more efficient. Moreover, the quality of the map provided by our scan matching method is good enough for a rescue robot.

Figure 2 provides an example of the maps generated by the implemented methods. The first image shows the odometry error of the robot in the environment; Figure 2b shows the results of our scan matching method; and Figure 2c shows the results of the Rao-Blackwellized algorithm. The range data are acquired through the combination of values coming from a LRF and a set of ultrasonic sensors. The figure also shows both that the Rao-Blackwellized algorithm is more accurate than scan matching and that this difference is acceptable for the task of returning a map of an environment by a rescue robot.

On the other hand, computational time for the scan matching method is significantly reduced. Typical cycle time for processing data on our robot is 10 ms for the scan matching method against 100 ms for Rao-Blackwellized algorithm.

V. NAVIGATION

In the past our team has competed in the RoboCup Soccer competition and successfully built teams of coordinating robots (we got the second place at RoboCup ’98) both for the Middle size and Legged Leagues. By partecipating in the Rescue competition we were given the chance to apply our experience and know-how to a much more challenging environment. Moreover, the Soccer domain is artificial and *a priori* known, while Rescue arenas are unknown, complex and unsafe environments.

For Soccer we have used a traditional approach for navigation. At each cycle the trajectory to a goal point was computed by a path planner [10], and a controller with visual feedback was used to follow it. This works well because: i) there were no obstructions in motion with the exception of other robots; ii) due to the high dynamics of the environment, long-term planning was not important for navigation; iii) the ability to negotiate narrow passages was not crucial.

In rescue arenas, however, we need more precision in the representation of the environment, because the robot must be able to navigate through narrow passages without colliding with obstacles. The size of the rescue area is not assumed to be known, so we had to change the path planner used in soccer because it wouldn't scale well; our new path planner uses instead a topological representation of the environment, built with a probabilistic approach ([4]).

Additionally, it is an any-time algorithm, therefore this has been the occasion to adapt the rest of the architecture to this type of algorithms; we are working on the possibility to have other any-time modules.

The advantage of any-time algorithms is that they provide a raw solution in a fast way, and this solution is refined with subsequent iterations; many of them have as disadvantages that it's generally not easy to estimate the quality of the current solution and sometimes it is not possible to acknowledge the non-existence of a solution in a finite time.

A. Path Planner

Our path planner builds a graph which reflects the environment topology thus reducing the path search in the entire map to a path search on a graph (roadmap method). Graph path will be subsequently deformed to make it executable by the controller and to make it easier to follow by the robot (for example it becomes smoother).

Map is built while exploring and using noisy sensor readings, thus resulting, in general, in a different map each cycle. This fact and moreover the resolution needed to deal with narrow passages, frequently found in a rescue environment, makes it hard to use deterministic algorithms that can work in real-time, therefore our choice has been to build the topological graph using a probabilistic algorithm.

It derives from Latombe's Probabilistic Roadmaps [17] and from Fritzke's Growing Neural Gas [11] and has also other characteristics: it produces a graph that reflects environment topology, not only connectiveness, and it modifies over time (simplifying itself as needed) reflecting changes detected in environment representation (it's not built from scratch at each cycle); it uses a set of heuristics that contribute to speed up the probabilistic process of building the graph. In this way, at any time, the path computation reduces to a path search on a *very simple* graph.

B. Controller

The controller module performs the task to make the robot follow the path computed by the path planner. In our case it implements a simple point stabilization using a linear interpolation along the path.

The problem with this kind of control is that it "blindly" follows the path, considering only estimated position as feedback. It doesn't deal with the possibility that some movement could unexpectedly fail and this occurs often in a rescue environment, for example when there are obstacles not detected by sensors or due to the non-planar and rough terrain. This results in the controller going on trying to make the same movement, because the robot remains unexpectedly in the same position.

VI. Exploration

Our exploration algorithm (derived from [30]) works as follows:

1) consider the frontiers between unexplored space and free explored space
2) choose one of them using a quality function
3) use the navigation module to reach it
4) go to 1 on goal reached or on timeout

The quality function can use heuristics like:

- the distance from the robot in the configuration space (i.e. nearer frontiers will be visited earlier, and those in front of the robot will be chosen before those behind);
- the distance from the boundaries of the area to explore (frontiers nearer to the boundaries will be visited first).

In a cell decomposition based algorithm we have to deal with the problem that some cell may contain an environment more complex than the others. Therefore the cells can be only partially explored and partially unknown. This causes a subsequent subdivision, resulting in a hierarchical cell decomposition of the environment. Hence our focus, in exploration task, is on unexplored areas of the map, considering frontiers allow us to take into account not only just those areas, without splitting them where it's not necessary, but also how we can reach them.

A. Results

We have successfully used our autonomous exploration module in a simulated environment (i.e. Stage). In this case exploration, completely autonomous, is finished in about 10 minutes, giving the full map of our real arena (modeled in the simulation system), which is similar to yellow rescue arenas, but without glasses and narrow passages.

VII. Human Body Detection through Stereo Vision

Semi autonomous victim detection in rescue scenarios is one of the tasks that we used in our system. This task is able to detect and find victims location by using the visual information and then add the position of the detected victim on the map. By considering the importance of victim detection and for avoiding false signalizations, the task asks for a confirmation before submitting the position of the victims. For victim detection we use a three step visual process based on stereo vision.

The three steps for human body detection are as follow.

- Segmentation and contour extraction
- Object classification
- Human body Modeling and Recognition

A. Segmentation and Contour extraction

Segmentation of images is the process of finding the bounds of each object in the scene by contour detection. Segmentation methods could be categorized into four classes: edge-based, clustering-based, region-based, split-merge based approaches. In a rescue environment, the

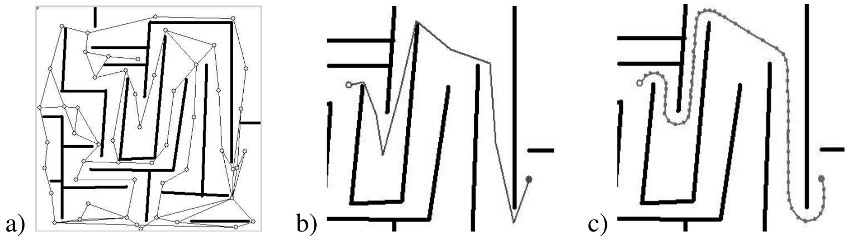

Fig. 3. a) The topological graph b) A path found in that graph c) The refined path, which will be sent to the controller

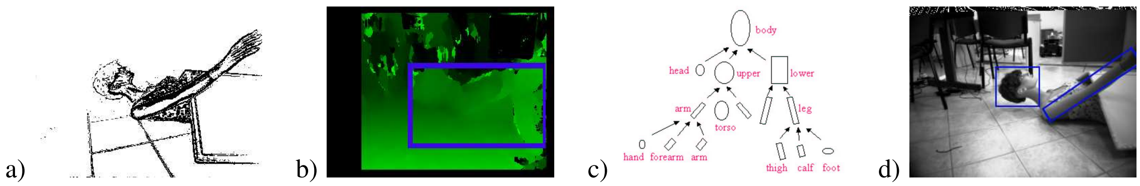

Fig. 4. a) Segmented image , b)Disparity image, c) Model, d) candidate objects

segmentation phase is very delicate, since the signal is disturbed by the noise and normally the bodies are covered by external materials. In order to implement an effective segmentation method for rescue robots, it is necessary to take into account the following guidelines: *extracting continuous contours, avoiding over segmentation, managing thresholds* . In addition to the above fundamental characteristics, an ideal method should require a computational time that makes it suitable for real-time applications. Among the methods for image segmentation, we believe that rescue robots may take advantage of clustering-based segmentation methods, because the contours of the regions are typically continuous. Over segmentation is resolvable by merging the homogeneous segmented areas after segmentation phase, also with this method we don't need to use any thresholds. As we use the stereo vision system we apply the segmentation only on one of the two images.

B. Object Classification

After the Image segmentation phase, it's necessary to identify significant objects in the segmented area. There are several methods that can be used for object classification but only a few of them are precise enough to classify human body parts.

In order to classify and object within a class of homogeneous objects at the first step it's important to have a good object class model. A good object model should allow the recognition of objects, independently from their position, orientation, size. For articulated objects (like human figures) this problem is obviously more difficult. It should also accommodate variations among the instances of an object class and should be insensitive to objects with partially missing parts. Human modeling is an essential part of model-based human detection. Although a great number of human models have been proposed in the literature, few of them are appropriate for human detection. Most models are developed for other purposes, such as human tracking[28], [12] or figure animation[16]. These models are either too complicated to be practical for efficient human detection, or can just be used to detect a particular person rather than all instances of humans. The common drawbacks with previous human models are:

1) the representations of human shapes are not invariant to similarity transforms, thus, they can only detect people of a fixed size or orientation;
2) the models are usually specific to a particular person, and do not model the statistical variance among individuals;
3) most models only represent the shape of a human body, but cannot handle the shape variation due to clothing;
4) although some models such as deformable templates can handle certain global shape variance [19], they have difficulty dealing with large articulated motion and partial occlusion.

The following list is the requirements of a good object class model for object classification[18]:

1) Not depending on scale, orientation, and position of objects;
2) Handles view-dependent shape variation;
3) Robust to shape distortions resulting from digitization noise and foreground/ background segmentation errors;
4) Robust to partial occlusions of an object;
5) Allow for articulated moving parts;
6) Influenced by the shape variations allowed within the class;
7) Support efficient shape recognition/ classification.

There is a large collection of literature on human modeling[18]. In this paper we categorize these methods in six categories:

- *part-based representation*
  There are many part-based representations to handle articulation. They vary widely in their level of detail.

At one extreme are methods that crudely model the body as a collection of articulated planar patches [28]. At the other extreme are 3D models in which the limb shapes are deformable [16], [12].

- *Cylinder and Super quadric-based methods:*[23], [22] Bowden et al.[26] encapsulated the correlation between 2D image data and 3D skeleton pose in a hybrid 2D-3D model trained on real life examples. The model they used allows 3D inference from 2D data. The common drawback with the above models is that they do not model the statistical variation among individuals and the effects of clothes on human shape. Thus, they may be used for human tracking or figure animation, but they are not appropriate for detecting people of various shapes and clothing also, their method does not generalize easily to new camera positions, because their 2D model is not invariant to viewpoint. As the objective of human body detection is to is to individuate the human body in any shape, form, color and size this method can not cover all of the needs for rescue missions.

- *Hierarchical based methods*
Marr and Nishihara[24] proposed a hierarchical 3D human model. At the highest level of the hierarchy, the body is modeled as a large extended cylinder, which is then resolved into small cylinders forming limbs and torso, and so on to fingers and toes. This hierarchical representation is stable in the presence of noise and sensitive to fine-level features, but it is often impractical since it contains a few constraints to support human detection.

- *Contour Based methods*
Contour-based representations have been used to model the 2D human shape. Baumberg et al. and Sullivan et al[2], [25]. employed a deformable template to handle shape deformation, where the shape model is derived from a set of training shapes. The orthogonal shape parameters are estimated using Principle Component Analysis (PCA). One drawback with this approach is that the model and the extracted contour should be aligned first, which is not a trivial task. Another drawback is that some invalid shapes are produced by the combination of two or more linear deformations. Gavrila et al.[7] Developed a template hierarchy to capture the variety of human shapes, and the model contains no invalid shapes. The common drawback with the above approaches is that they do not model individual parts, and so they can only handle limited shape variety due to articulation and cannot deal with occlusion very well.

- *Skeleton-based methods*
Skeleton-based representations[3] have been used to model the topological structure of the human body, but they do not model the shapes of body parts. These approaches are sensitive to noise as a very active factor in rescue scenes and cannot distinguish two classes with the same topological structure but different geometrical structures.

- *Other methods*
Some models incorporate other cues or features into the model. Pentland[6] introduced a blob-based representation that combines skin color and contour to represent a body part. While the color-blob representation of a person is quite useful, it is not invariant under clothing/lighting changes and so it requires an initial model learning procedure for different subjects and a smoothly changing image background. Papageorgiou et al[8]. developed a wavelet-based representation to model pedestrians, but this representation is not invariant under rotation and can not handle large part movements and occlusion very well.

In our System we use a Skeleton based method, By using the Stereo Calculation we can apply the the third dimension (Z) to the segmented object (where it is possible to compute), and compare the object with our Global model of the human body as a Matrix of the relations and Joints.

C. Human body Modeling and Recognition

The third step is the joining and reconstructing of the detect part to find the current position of the body, in our method there are a strict interaction between classification and modeling. There are several methods to detect the composition of the body. The problem of determining the similarity of two shapes has been well studied in several fields. The design of a similarity measure depends on how a shape is presented. Some categories of these methods are: *global shape description, point based similarity, part-based representation*.

In our system we use the classifier and the Body modeler in recursive mode, it means that each time that the classifier match a segmented object to the models database, the modeler controls if the position of the object is correct respect to the global body model representation Figure 4, Figure 5.

D. Experimental Results

We used the Victim Detection System in the RoboCup-Real-Rescue 2004 competitions at Lisbon(Portugal). In the six rounds the system found 4 victims in autonomous mode. With the current set up and body model/joints matrix the system is able to detect the victims in the surface or partially trapped situation, in the partially trapped situation at least the position of the torso or the legs should be visible. The body part model is trained by a set of 200 stereo images.

VIII. Conclusions and Future Works

Our system allows a full autonomous mapping and localization and a semi-autonomous victim detection. This speeds up the whole process of victims searching. Once

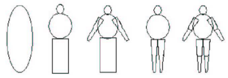

Fig. 5. Some candidate models from the model database

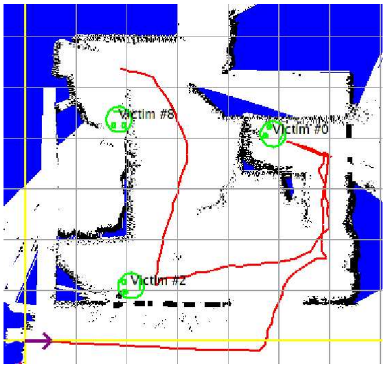

Fig. 6. The map of the Yellow Arena in RoboCupRescue Real at Lisbon 2004, showing also victims locations and the path done by the robot.

solved navigation issues, we will have the opportunity to use multiple heterogeneous robots, controlled by a singel operator. This for increasing the exploration speed and having more chances for navigating.

We are investigating to extend our algorithm in order to make them even more useful in rescue scenarios.

A. Mapping and localization

For what concerns mapping task, we are analyzing mapping methods for non-planar environments. When the robot that maps the area does not move on a plane, the problem of SLAM becomes more difficult, for the following reasons:

- The robot pose space passes from from three to six dimensions. If we want to track the robot pose with a Rao-Blackwellized particle filter, we need a cubic number of samples to have the same coverage of the pose space.
- If we are using a wheeled robot, the odometry sensors provides a bad estimate when moving on a non regular surface.
- The laser range finder, that can be safely used in planar environments loses its effectiveness, since it is able only to detect obstacles lying on the scanning plane, therefore we must switch to more inaccurate sensing device like stereo cameras. Moreover, as far as the laser can not be used all of the effective scan-match based approaches cannot be used.

For the above reasons the most proper sensor equipment for a 3D mapping robot includes an accelerometer, for dead reckoning, a compass, and a stereo camera as eteroceptive device. Moreover, since the stereo camera provides sparse and noisy informations, it is extremely difficult to apply some dense sensor matching technique (like scan match), and it is needed to consider feature based SLAM techniques. In order to provide the operator with a feasible environmental representation for operation, a reconstruction technique can be used on the path estimated by the SLAM algorithm.

The state transition is governed by the odometry, the accelerometer, and the compass. At each time a local view of the landmarks is built from the estimated pose and the map, and the data association is solved using the nearest neighbor principle as in many SLAM approaches. To increase the robustness of the approach with respect to association failures FastSLAM can be effectively used, provided that the sample space dimension has been reduced, by the use of the compass and the accelerometer.

We are currently performing a deeper analysis and more experiments in order to devise a suitable configuration and the appropriate technique that can be effectively used in non-planar environments.

We are also trying a completely different approach to scan match using the Hough transform for Planes ([5]).

B. Navigation

In navigation, the main issues which we are analyzing is to take in consideration:

- the true shape of the robot and the sometimes complex manoeuvres needed to deal with narrow passages;
- controller fails and a strategy which can make the robot escape from stall situations, or in general to avoid to repeat a failed move.

In order to take into account the real shape of the robot, we have to extend the topological graph building process, adding the ability to deal with narrow passages, in which only a subset of movement can be done, and the robot could not rotate, for example, by any degree.

On the other hand, we will try to detected and avoid movements and areas which often causes controller to stall (e.g. rotations with 0-radius angle), for example recording critical locations (e.g. which previously caused a stall) in the map and eventually make the path planner avoid them. The controller, moreover, has to be ”smarter” and detect if a movement has failed and try another one (but also going on following the path). This could lead to add another module which have to communicate with controller and path planner.

C. Exploration

Given a set of navigation modules capable of moving the robot even in narrow passages and in rough and hard terrains, where the controller could fail to do some movement, our exploration works well without great changes, as demonstrated in experiments in simulated environments.

Anyway, we are tuning up quality function parameters and try to use a more sophisticated planner, in order to avoid unuseful moves and so speed up exploration process. Another interesting issue could be to integrate exploration decisions with the human body detection module, thus better exploring areas in which victims seem to be present.

D. Victim Detection

The Current Method for Victim detection is sensible to light conditions, we are working on the methods of segmentation that have less sensibility to this variations. Another aspect that we consider is integrate the other sensors (Infra Red, Acoustic, Gas) to the algorithm(in the current state the operator receive this informations directly from the sensors and they don't take any effect on the method).

Also we are working on the self/external body occlusion by making a more stable model for the body.

REFERENCES

[1] Carmen project, `www-2.cs.cmu.edu/ carmen/`.

[2] D. Hogg A. Baumberg. Learning flixable models for image sequences. *Proc. Europian Conf. on Computer Vision*, 1994.

[3] S.C.Zhu andA.L.Yuille. Forms: A flexble object recognition and modeling system. 1996.

[4] D. Calisi. Pianificazione del percorso per un robot tramite mappe topologiche probabilistiche e adattamento della traiettoria (in italian).

[5] A. Censi, L. Iocchi, and G. Grisetti. A new approach to scan matching using the hough transform. Submitted to ICRA05.

[6] Pentland C.Wern, Azarbayejani. Pfinder:real time tracking of the human body. 1997.

[7] D.M.Gavrila and Philomin. Real-time object detection for smart vehicles. *Int'l Conf.On Computer Vision*, 1999.

[8] C.Papageorgiou et al. A trainable pedestrian detection system. 1998.

[9] A. Farinelli, G. Grisetti, and L. Iocchi. Spqr-rdk: a modular framework for programming mobile robots. In *Proc. of Int. RoboCup Symposium*, 2004.

[10] A. Farinelli and L. Iocchi. Planning trajectories in dynamic environments using a gradient method. In *Proc. of the International RoboCup Symposium 2003*, Padua, Italy, 2003.

[11] Bernd Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA, 1995.

[12] D. M. Gavrila and L. S. Davis. 3-d model based tracking of humans in action: A multi-view approach. *CVPR'96*, pages 73–80, 1996.

[13] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. Submitted to ICRA05.

[14] J.S. Gutmann and K Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE Interational Symposium on Computational Intellignece in Robotics and Automation*, 2000.

[15] D. Hanel, W. Burgard, D. Fox, and S. Thrun. An efficient algorithm for generating maps of large scale cyclic environments from raw laser range measurements, 2003.

[16] D. Metaxas I. A. Kakadiaris. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. *CVPR'96*, pages 81–87, 1996.

[17] L. Kavraki and J. Latombe. Probabilistic roadmaps for robot path planning, 1998.

[18] liang Zhao. *Dressed human modeling, detection, and parts localization.* PhD thesis, The Robotic Institute Carnegie Mellon University,Pittsburgh, 2001.

[19] T-L. Liu and D. Geiger. Approximate tree matching and shape similarity. *Proc. Intl. Conf. on Computer Vision, Kerkyra, Greece*, 1999.

[20] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *CVPR94*, pages 935–938, 1994.

[21] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping, 1997.

[22] A. Sato et al. M. Yamamoto. Incremental tracking of human actions frommultiple view. *CVPR98*, 1998.

[23] S. Kawada et al. M. Yamamoto, A. Sato. Tracking persons inmonocular image sequence. *Computer Vision and Image Understanding, Vol. 7*, 1999.

[24] D. Marr and H.K. Nishihar. Representation and recognition of the spatial organization of three-dimensional shape. *Proc. Roy. Soc. B, B-20*, 1977.

[25] etal. M.Sullivan, C.Richards. Pedesterian tracking from stationary camera using active deformable models. *Proc.Intelligent Vehicles*, 1995.

[26] M. Sarhadi R. Bowden, T. A. Mitchell. Reconstructing 3d pose and motion from a single camera view. *BMVC'98*, 1998.

[27] Ioannis M. Rekleitis, Gregory Dudek, and Evangelos E. Milios. Probabilistic cooperative localization and mapping in practice. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-03)*, LasVegas, USA, 2003.

[28] Y. Yacoob S. Ju, M. Black. Cardboard people: A parameterized model of articulated image motion. *Proc. Int. Conf. on Automatic Face andvGesture Recognition*, pages 38–44, 1996.

[29] C. Stachniss, G. Grisetti, D. Haehnel, and W. Burgard. Improved rao-blackwellized mapping by adaptive sampling and active loop-closure. In *In Proc. of the Workshop on Self-Organization of AdaptiVE behavior (SOAVE)*, 2004.

[30] B. Yamauchi. A frontier based approach for autonomous exploration, 1997.