

An evolutionary strategy for achieving autonomous navigation

Douglas W. Gage

Space and Naval Warfare Systems Center San Diego
SPAWARSYSCEN D371, San Diego, CA 92152-7383

ABSTRACT

An approach is presented for the evolutionary development of supervised autonomous navigation capabilities for small "backpackable" ground robots, in the context of a DARPA-sponsored program to provide robotic support to small units of dismounted warfighters. This development approach relies on the implementation of a baseline visual servoing navigation capability, including tools to support operator oversight and override, which is then enhanced with semantically referenced commands and a mission scripting structure. As current and future machine perception techniques are able to automatically designate visual servoing goal points, this approach should provide a natural evolutionary pathway to higher levels of autonomous operation and reduced requirements for operator intervention.

Keywords: mobile robot, autonomous navigation, visual servoing

1. INTRODUCTION

The development of autonomous mobile robots with non-trivial navigation capabilities began as an interesting application domain for Artificial Intelligence researchers in the late 1960s, and it continues to present major challenges to researchers and system developers today.¹ While unmanned ground vehicles have demonstrated some level of competence in both on- and off-road autonomous driving, the goals set for DARPA's Autonomous Land Vehicle (ALV) program in the middle 1980s have not yet been fully met. The successful exploration of a very limited area of the Martian surface by JPL's Sojourner rover in 1997 was made possible only by an intense level of operator supervision.² A number of programs have been undertaken in recent years to develop mobile robot systems for a variety of real-world applications such as site security³ and unexploded ordnance (UXO) disposal⁴; this paper presents a strategy for developing enhanced supervised autonomous navigation capabilities for one of these programs, Tactical Mobile Robotics.

1.1 DARPA TTO's Tactical Mobile Robotics (TMR) program

Tactical Mobile Robotics (TMR) is a technology development program undertaken by DARPA's Tactical Technologies Office (TTO) to develop a system of small robots that can be deployed and used by dismounted warfighters in urban operations. While specific system requirements continue to be refined by the TMR program, the nominal baseline conceptual requirements for such a robot are:

- TMR robots must be capable of being transported, deployed, and controlled by dismounted warfighters at the platoon or squad level. This implies a maximum size while being transported of about 60 cm by 50 cm by 20 cm (24" by 20" by 8"), and a maximum weight of about 10 kg (20-25 pounds). While system modularity is highly desirable to support a variety of mission payloads, a warfighter must be able to configure and deploy a robot in bitter cold and complete darkness.
- TMR robots must be able to effectively and rapidly negotiate urban terrain: to clamber over rubble, to climb up and down stairs, and, ideally, to open and close doors.
- TMR robots must maintain constant communications with the operator and with each other, even in the communications-unfriendly urban environment.
- TMR robots must be easy to use, be able to navigate based on a minimum level of intuitive operator direction, and not get lost.

- The TMR system must accept and require an appropriate level of operator interaction in a number of very different situations. The system must allow the operator to assert precise control when desired, but should offer no distraction to warfighters in the heat of battle, and must not get in the way or slow the tempo of operations.

The TMR Program is planned as a four year program in two two-year phases. The first phase, which began in June 1998, includes (1) technology development contractors working (under DARPA BAA 98-08 part A) in various areas of sensors (University of Michigan), perception (Yale University and SRI International), autonomy (SRI International, Carnegie Mellon University, Stanford University, University of Southern California, and Georgia Tech), and mission packages (Foster-Miller), and (2) three parallel system design contractor teams (under DARPA BAA 98-08 part B), led by Draper Laboratory, Raytheon Systems, and SAIC, who are also addressing additional systems requirements such as power sources, communications, and the operator interface. The second phase of the program will pursue the implementation of one or more of the system design concepts developed in the first phase, leading to a final demonstration in mid-2002.^{5, 6}

The second phase of work on DARPA TTO's Urban Robot program (DARPA BAA 97-20) was folded into the TMR Program in mid-1998. Several first phase 97-20 performers combined to form a team tasked to provide an initial demonstration of TMR capabilities in the summer of 1999. This team is led by NASA's Jet Propulsion Laboratory (responsible for high level sensors and electronics, software architecture, and system integration, as well as for project management), and also includes Carnegie Mellon University (driving modes and a panoramic sensor), IS Robotics (mobility platform, plus low level sensors and associated electronics and behaviors), Oak Ridge National Laboratory (indoor localization, mapping, and path planning), and University of Southern California (operator control unit).

Three navigational (driving) modes will be implemented for the 1999 demo: (1) waypoint navigation, which drives the vehicle through a sequence of locations which the user has designated in an image, (2) map based path planning, which uses sensor-based position information to drive to a location specified by the user on an a priori map, and (3) visual servoing, which drives the vehicle to the location of a goal-point template which the user has designated in an image. This third capability, visual servoing, serves as the baseline for the evolutionary development strategy for autonomous navigation presented below.

1.2 Challenges to realizing TMR

A number of key challenges must be met in order to successfully develop a capable TMR system. For example, the size constraint on TMR robots presents two critical challenges. The first is simply to be able to do the required job; to be able, for example: (1) to negotiate (as opposed to merely avoid) obstacles of a size comparable to the robots themselves, (2) to rapidly climb up and down stairs, (3) to open a door and hold it open while passing through the doorway, and (4) to read human-eye-height hotel room numbers in a narrow corridor. Clever mechanical design will be required to achieve these capabilities with a small robot. The second size challenge is in implementation: a straightforward COTS ("Commercial Off The Shelf") hardware implementation of the required sensor and processing functionality and performance will not physically fit in the desired TMR envelope. Given TMR's limited program resources, success will depend on making smart decisions as to which components to miniaturize, and in cleverly integrating these elements with a minimum of chassis, cables, and connectors. (At the same time we must also ensure that our integration architecture is flexible enough to allow us to incorporate improved subsystem technologies as they become available.) Timing is also important: technical and budget risk is almost always reduced by delaying miniaturization, but TMR must demonstrate real capabilities as early as possible in order to establish and maintain a strong user constituency.

Another challenge is to focus TMR resources on work that the TMR program must actually accomplish itself. The TMR strategy is to define the overall system so as to minimize the number of critical "voids" that must be satisfied in order to provide an initial kernel of effective and user-acceptable capabilities. Furthermore, TMR makes maximum leverage of other developments in robotics (e.g., the Demo III program), perception (e.g., DARPA ISO's Image Understanding Program), user interface (e.g., DARPA ETO technologies), power sources (DARPA TTO technologies), and communications (rapidly evolving COTS technologies, as well as numerous programs across the Department of Defense), so that we can focus TMR's resources to address those remaining voids that are TMR-specific. The TMR technology contracts provide the vehicle to fill the critical TMR-specific "voids" in subsystem level functional and performance capability that are required to realize the TMR system concept.

One problem in trying to leverage previously developed technologies is that subsystem level (e.g., machine vision or obstacle avoidance) research results do not necessarily represent real capabilities implemented on or immediately transferable to real robots. A "capability" may have been demonstrated only in a single specific environment, or even only in simulation (and experience has shown that the performance of perception-based navigation schemes can be remarkably dependent on the details of the environment, so that extensive testing in a variety of locations is an absolute necessity⁷). It is necessary to carefully evaluate a technology's maturity and applicability to a real robotic system that must be demonstrated in just a few years. In several cases -- visual servoing for ground vehicle navigation (as opposed to manipulator control), perception-based retrotraverse (as opposed to retrotraverse based on a precision localization technique like differential GPS), and detecting and reading the text of signs (e.g., traffic signs or street signs) -- it appears that the research community has not yet implemented anything like a "COTS" capability, not because of the difficulties involved, but because these are not considered attractive research issues.

In summary, the success of TMR will in large measure depend on clever system/product conceptualization, adoption of an effective and flexible architecture at both the TMR system and robot-platform levels, the timely maturation of a number of key subsystem technologies, the diligent execution of the overall system integration process, and an extensive program to assess and validate system performance in a wide variety of real-world environments.

1.3. TMR navigation and operator tasking

Perhaps the single area in which the TMR system conceptual design can help make or break the program is in deciding just how a human operator will direct a robot where to go, and how the robot will actually accomplish getting there.

Every human naturally acquires in early childhood (and thereafter exercises virtually unconsciously) the skills needed to successfully navigate in the real world, i.e., to: (1) move, (2) avoid bumping into anything, (3) understand where he or she is trying to go, and (4) figure out how to get there. Each of these capabilities, however, must be explicitly implemented by the developer of a mobile robot: (1) locomotion (or mobility), (2) obstacle avoidance, (3) global navigation, and (4) path planning.

Moreover, a human easily interprets the ambiguities of natural language, and is naturally able to detect, classify, and identify specific environmental features under widely varying environmental conditions, and independent of relative orientation and distance. So a human can easily understand and execute a task presented as:

"Go down this road about a mile and turn left on Union Street -- it's the second or third light, I think -- and then turn right into the alley just past the McDonald's; it's the second house on the left, the green one with an elm tree in front - - you can't miss it."

A robot can't do this. Besides not being able to understand natural language, current robots are emphatically NOT "able to detect, classify, and identify specific environmental features under widely varying environmental conditions, and independent of relative orientation and distance".

Teleoperation, in which a remote human operator provides continuous driving inputs while watching video returned from the vehicle, has over the years supported the fielding of numerous unmanned systems in such application areas as bomb disposal and nuclear reactor maintenance. While teleoperation will also play a role in TMR -- especially safeguarded teleoperation, in which onboard sensors prevent collisions with obstacles while the remote driver sets the general direction of travel -- it will seldom be acceptable for TMR to require continuous operator attention. The goal for TMR is to implement a mode of navigation which can provide a real increment of autonomous operation (reducing the load on the operator) without requiring significant advances in machine perception technologies.

2. A NAVIGATION STRATEGY BASED ON VISUAL SERVOING

TMR's development strategy for realizing autonomous navigation capabilities is based on the observation that machine perception is the limiting factor in robotic autonomy, and the assumption that this fact is unlikely to change in the near future. As a consequence, the goal is to accept a requirement for a moderate level of operator input in order to maximize the system's level of navigational performance for a given level of machine perception capability, even as the TMR technology contractors work to improve these capabilities. The development strategy includes the following components:

- Implement a baseline visual servoing capability, by which the robot moves toward a goal point (visual template) designated by the operator in an image returned by the robot, using the robot's organic obstacle avoidance capabilities.
- Supplement visual servoing with tools for operator oversight and override, so that the operator can monitor system performance and designate a new goal point if the robot gets "lost" or if mission requirements change.
- Add a degree of semantic reference in terms of intent and environment to the command structure -- in other words, have the operator tell the system that the goal point represents a vehicle or a door or a bush. This can provide a crutch for the robot's limited perception capabilities and also cue system responses appropriate to the situation.
- Incorporate multiple semantically referenced visual servoing commands into a mission scripting structure, to support advance mission planning and to reduce operator load during mission execution.
- As feasible, exploit current and future machine perception techniques to automatically designate visual servoing goal points, while retaining human operator oversight.

2.1 Visual servoing

The term "visual servoing" originated in reference to machine vision based closed-loop position control for an industrial robotic manipulator's end effector, and most work in visual servoing has addressed manipulator control.⁸ Applied to mobile robot navigation, visual servoing essentially supports a command of the form "GOTO <goal point in image>".

Operator tasking. Initially, the tasking image is a "current" image, i.e., an image taken from the current position of the robot, taken with the camera that will support the actual execution of the command, and taken recently enough that lighting and other factors have not changed. (A useful extension of this capability would allow the use of images taken with other cameras, from somewhat different positions, and with different lighting and weather conditions.) The designated goal point must clearly correspond to some physical location, so a point "on" the left "edge" of a cylindrical building would clearly not work, nor would a rainbow. Moreover, the designated point must serve to identify some visual template (intersection of edges, or "blob") in the image that will be "tracked" in succeeding images.

Task execution. "GOTO" refers to the actual process of moving to the goal point. This capability can obviously be more or less powerful and robust, at minimum being limited to traversal of an unobstructed straight line trajectory to the goal point, then adding (1) competences to deal with various degrees and types of non-traversability (obstacle avoidance or negotiation), and (2) the ability to reacquire the goal feature in the image if it is "lost" during the traversal. "Pure" visual servoing uses only the 2 dimensional image; the process may also make use of 3 dimensional information (i.e., the distance to the goal), and/or be supported by some local and/or global mapping capability.

Operator oversight and override. While the GOTO process is executing, the operator is periodically provided with status information in the form of later images with the goal template annotated, so that if the robot gets lost and doesn't realize it, the operator can detect that fact and intervene. The operator prescribes the frequency of these reports. In addition, other data may be made available to the operator, such as the current estimated distance to the goal, estimated time until arrival, and some measure of the quality of goal tracking (e.g., how many times has the visual goal feature been "lost" and reacquired).

Termination conditions. If the operator does not intervene, then one of several outcomes eventually ensues:

- The process normally terminates when the robot reports that it has arrived "at" the goal point. The "at" criterion, which is specified as a parameter of the command, determines how close the robot must be to the goal before it declares success and prepares to execute its next command.
- The process terminates when the robot decides that it has irrevocably lost its visual track of the goal template. The criteria for this decision to give up and admit defeat, in terms of time and/or distance traveled without visually acquiring the goal, are also specified as parameters of the command.

- Since the command must terminate even if, for example, the robot somehow manages to generate a closed loop in its trajectory, a timeout termination condition must also be specified.
- Yet another command parameter allows the operator to specify how good the match has to be -- how confident do we demand that the robot be in its tracking (or reacquisition) of the goal template.

Given these additional considerations, it is seen that the full specification of a visual servoing command includes a number of additional parameters, either explicitly or implicitly: "GOTO <goal point in image> returning <how much> status <how often>, matching the goal feature <how well>, and terminating <how far from the goal>, or when <failure condition>, or after <timeout period>."

2.2 Semantic references

Adding semantic references to the navigational command structure allows the operator to explicitly inform the system about the intent of the command and specific relevant features of the environment. Knowing the characteristics of the physical object represented by the visual goal template can make it easier for the system to visually track it, and the command semantics can also provide a concise way of specifying desired behaviors -- e.g., "this is a bush; wiggle around so you can hide in it."

It is important to stress that this requires the recognition of only a limited number of object types (semantic categories), which are chosen because they are both (1) relevant to command- and mission-level execution and (2) amenable to some degree of automatic detection and classification using current perception processing techniques and sensed parameters such as size, shape, texture, color, motion, and IR and audio signatures. TMR's initial set of semantic categories includes humans, animals (dogs), vehicles, surfaces (pavement, earth, and grass), roads, vegetation (brush, shrubs, and trees), walls, hallways, doorways, doors, windows, and stairs.

This approach of basing navigation function on semantic reference is exactly what is done in well-known indoor navigation schemes based on ultrasonic sensors, such as wall- or hall-following,⁹ and in vision-based road following,¹⁰ and these capabilities will in fact also be implemented on TMR.

Example Command: Move Under <this> Vehicle. The designated region of the image includes a vehicle, and the software performs a check to verify this. Using visual servoing (whose goal-tracking performance may be enhanced by knowing that the goal template is a vehicle), the robot moves to the vehicle. When it arrives at the vehicle, it attempts to hide itself beneath it. In this process it seeks to avoid getting stuck, and it expects to lose both GPS satellite fixes and communications links. While underneath the vehicle, follow-on commands could tell the robot to jockey about in order to try to reestablish communications and/or to reacquire GPS, and/or to listen for the starting of the sheltering vehicle's engine so that the robot can go hide under another nearby vehicle if this one prepares to move away.

Example Command: Climb Up <these> Stairs <how high>. The designated region of the image includes the base of a flight of stairs, and the software performs a check to verify this. The command also specifies how high up the stairs the robot should go, and whatever a priori knowledge is available about the stairway parameters (tread depth, riser height, steps per floor, landing and stairwell configuration, etc). Using visual servoing, the robot moves to the base of the stairs. When it arrives there, it uses its sensors to determine (or verify) the stairway parameters, positions itself, and then climbs the stairs, continuing to refine its estimate of the parameters as it climbs.

Other commands include:

- Climb Down <these> Stairs
- Cross <this> Street (and avoid collisions while crossing)
- Hide in <this> Vegetation
- Move Along <this> Wall (until...)
- Move Down <this> Hall (until...)
- Open <this> Door (and Enter... and Close)

- Move in <this> Direction (until...)
- Wait until... (humans are (not) present...)

2.3 Mission scripting

The TMR system can accept and execute a mission script -- a pre-specified sequence of detailed commands which, as they execute, prompt for operator input (to initiate command execution or to designate targets) and accept operator override (to correct mistakes or to change the overall plan). The system must therefore provide a user-friendly script editing capability.

Writing an effective script requires both (1) detailed knowledge of the operational environment and (2) time and effort from a well-trained operator. Different operational environments (i.e., indoor, urban, rural) will tend to favor the use of different subsets of the command set, and achieving optimum performance will require careful tuning of the command parameters (in other words, "style counts"). Rehearsal, when possible, will be extremely valuable for script refinement and validation. Mission scripting capability will thus provide its greatest operational payoff when enough detailed information is available far enough in advance to support careful planning. In military parlance, this means that scripting is most valuable for *deliberate* operations, vs *hasty* or *battle drill* situations.

2.4 Evolutionary development

While the goal points for visual servoing-based commands are initially provided by the operator, it is only the limits of its perceptual capabilities that prevent the system from autonomously acquiring many goal points from its sensor inputs. Thus, an additional designation mode for goal-oriented commands will take the form "GOAL = find <type of object> in bearing range <leftmost, rightmost> and distance range <nearest, farthest>". The image, with the designated GOAL annotated, will be presented to the human operator, who can either (1) override the designation or (2) permit the command to execute using the autonomously designated goal. This mode will provide a useful operational capability consistent with system perception capabilities, and provide a base of experience that can help focus the evolution of those capabilities ("Damn! This thing works fine finding bushes -- except for the !#%\$* bushes around here.")

The implementation of the command and scripting capability must be as flexible as possible, to facilitate the continuing evolution of the command structure. Commands should be interpreted, rather than compiled, to enable the rapid modification (tailoring) of individual commands and the extension of the command set by creating command macros.

3. PATH-REFERENCED NAVIGATION MODES

A second class of navigation behaviors that will play an important role in TMR are those whose tasking and execution are referenced to the paths that TMR robots have previously traveled in the course of the mission, rather than to imagery or maps. (These might be termed "Been There, Done That" modes.) The execution of these behaviors makes use of localization data (such as DGPS waypoints or the characteristics and spatial relationships of sensed "landmarks") acquired during a robot's movement (whether autonomous or teleoperated) to allow any of the robots to autonomously follow the same path a second time, or in the reverse direction.

Examples of behaviors in this class are:

Platooning/convoying: multiple robots travel as a group along the same path without interfering with each other. This capability can be implemented in a number of different ways, one of which is "follow the leader", in which case the user could teleoperate the leader robot, and the other robots would follow automatically.

Route replay: a robot is directed to autonomously travel the same path that another robot has previously traveled. While similar to convoying, this capability differs in that its successful execution cannot depend on the physical presence of other robots.

Retrotraverse: a robot is directed to retrace its steps. Sensor directionality (forward-pointing versus backward-pointing) must be accounted for in implementing the retrotraverse capability.

"Go to <this> previously visited location": a robot is directed to revisit a specific location that it (or another robot) has previously visited. Execution of this task may involve splicing together segments of route replay and retrotraverse.

The purpose of these path-referenced behaviors is to make maximum leverage of limited perception capabilities in order to minimize the level of operator attention required in both the tasking and execution of repeated specific tasks. The TMR system memorizes the localization parameters generated by each robot as it traverses each path segment, whether teleoperated, fully autonomously, or with some intermediate degree of supervision. The system can later use this data to re-execute the same traverse (in either direction) in a fully autonomous mode. The system must of course also provide the user with an easy way of specifying the desired goal point in terms of map location, time of previous visit, image or other sensor data, or other labels.

These are truly system-level capabilities, and require that data be stored at the system level, and not merely be retained in an individual robot. The challenge is to achieve the appropriate level(s) of abstraction in the data representations adopted. This data must be detailed enough to successfully support navigation, but must also be compact enough to avoid exceeding the TMR system's communications capacity.

Path-referenced navigational modes provide a big operational payoff because they allow the user to task the system in terms of mission level events. They therefore also allow TMR to satisfy a fundamental reasonable user expectation -- if you've already told the system the details of how to perform a certain specific task, you shouldn't have to tell it again.

4. MISSION-ORIENTED TASKS BASED ON AUTONOMOUS NAVIGATION

A third class of navigation functions applicable to TMR are higher-level mission-oriented tasks which make use of autonomous navigational capabilities to perform a function beyond simply navigating to a desired destination point.

Many such tasks involve area coverage: the mission is to carry some sensor or effector mission package on some desired (although not necessary predetermined) path through the operations area. In the civilian agriculture sector, DGPS-controlled tractors are now being used to plow, seed, fertilize, and harvest fields. Potential military applications include numerous variations on the theme of area search: search and rescue (SAR), minefield breaching, demining, and UXO disposal.

TMR involves at least two such navigation-based mission-oriented tasks: the adaptive maintenance of communications connectivity, and the mapping and monitoring of activity in building interiors.

4.1 Adaptive maintenance of communications connectivity.

The urban environment is very unfriendly to military RF communications. Building structures can both block and reflect RF signals, so that communications is unreliable outdoors, while indoors it is even more problematical. As dismounted units, individual troops, and TMR robots move through the urban environment, communications must be maintained between robots and their controllers, between maneuvering elements, and between these elements and higher headquarters. One key role which has been identified for TMR is that of "commsbots": autonomously repositioning communications relays.

In "commsbot" configuration, a TMR must carry the communications gear appropriate to the unit's mission, plus any additional hardware required to measure RF signal strength and/or direction, as well as extra batteries or other power sources adequate to support the desired mission profile. TMR commsbots collaboratively reposition themselves to maintain specified communications connectivities by monitoring current and evolving link connectivities and signal strengths and planning the coordinated movement of multiple robot relays to optimize overall system connectivity as the operation unfolds.

4.2 Mapping and monitoring of activity in building interiors.

A principal MOUT (Military Operations in Urban Terrain) function is that of "clearing" a building -- small teams of warfighters enter and methodically move through a building, room by room and floor by floor, neutralizing any threats encountered. While a deployment strategy involving robots actually accompanying troops into a building raises serious issues of interference and distraction of the troops' attention, a group of robots could be deployed in advance of the troops' entry into

a building in order to map its interior and to detect, localize, and identify threats. These robots would work collaboratively, moving in concert to ensure that threats are not able to evade detection as the robots move through the building.

While the goal in both the communications relay and building mapping applications is that actual robot movements should always be executed autonomously, the system can request operator advice and supervision as necessary.

5. CONCLUSION

Caveat: As described above in the introduction, the first phase of the TMR program began in June 1998, while the final demonstration of TMR system capabilities is not slated until the end of the second program phase, in mid-2002. This paper is therefore clearly not a description of completed work; instead it is a manifesto presenting program-level goals and a strategy for technology development in the critical area of navigation. It represents a snapshot of current thinking which is likely to evolve as the program unfolds.

The TMR development strategy is intended to allow the TMR program to demonstrate useful navigational capabilities to the dismounted warfighter user community without having to depend on any significant advances in machine perception. The representation of a robot's mission tasking as a script of semantically referenced commands carves up the overall TMR system requirement for "perception" into a set of "small" (in the sense of well-defined and bounded, but not necessarily easy) explicit perceptual tasks, each of which is then allocated as appropriate either to a machine perception competence or to the human operator. The operator's tasking interface (including the script editor) supports specification of what the robot is to do, where to do it, and when to do it. Oversight/override tools allow the operator to monitor task execution and to intervene when necessary. New improved machine perception competences can be incorporated into the system as they mature, without having to revise the scripting interface. Instead they will appear as new commands, smarter commands, and a reduced level of required operator override, reflecting an evolutionary reallocation of individual perception tasks from the operator to the system. In addition, path-referenced navigation modes, such as platooning and retrotraverse, and coordinated robot group behaviors for maintaining communications and performing reconnaissance inside buildings, will evolve to provide additional levels of autonomous capability requiring minimum user supervision.

ACKNOWLEDGEMENTS

This work has been performed under the Tactical Mobile Robotics (TMR) Program sponsored by the Defense Advanced Research Projects Agency (DARPA) Tactical Technologies Office (TTO). The author acknowledges numerous productive conversations with many participants in the TMR program, especially with Martial Hebert.

REFERENCES

1. D. W. Gage, "A Brief History of Unmanned Ground Vehicle (UGV) Development Efforts", **Unmanned Systems**, volume 13 no. 3, pp 9-32 summer 1995.
2. B. Landry, "Take a Deep Breath", <http://quest.arc.nasa.gov/mars/team/journals/landry/16.html>
3. H. R. Everett, "Breaking Down the Barriers," **Unmanned Vehicles**, Vol. 3, No. 1, pp. 18-20, February-April 1998.
4. C. DeBolt et al, "Basic UXO Gathering System (BUGS): Multiple Small Inexpensive Robots for Autonomous UXO Clearance", **Proceedings of Third International Symposium on Technology and the Mine Problem**, Monterey CA, 6-9 April 1998.
5. "Tactical Mobile Robotics (TMR)", <http://www.darpa.mil/tto/tmr.html>
6. "Tactical Mobile Robotics (TMR)", <http://www.spawar.navy.mil/robots/land/tmr/tmr.html>
7. H. R. Everett, D. W. Gage, G. A. Gilbreath, R. T. Laird, R. P. Smurlo, "Real-world Issues in Warehouse Navigation," **SPIE Mobile Robots IX**, Vol. 2352, Boston, MA, pp. 249-259, November, 1994.

8. S. Hutchinson, G. D. Hager, and P. I. Corke, "A Tutorial on Visual Servo Control", **IEEE Transactions on Robotics and Automation**, vol 12, no 5, pp 651-670, October 1996.
9. H. R. Everett, G. A. Gilbreath, and J. M. Holland, "Hybrid Navigational Control Scheme," **SPIE Mobile Robots V**, pp 291-298, November, 1990.
10. M. Hebert, D. Pomerleau, A. Stentz, and C. Thorpe, "Computer Vision for Navigation: the CMU UGV Project", **Proceedings of the Workshop on Vision for Robots**, pp 87-96, IEEE Computer Society Press, 1995.