REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
29 April 1996

3. REPORT TYPE AND DATES COVERED
Final Technical Progress Report

4. TITLE AND SUBTITLE
Vehicles for Selective Vision and Control

5. FUNDING NUMBERS
DAAH04-95-1-0050

6. AUTHOR(S)
Christopher M. Brown

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Rochester
Computer Science Department
734 Computer Studies Bldg.
Rochester, NY 14627-0226

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of the Army
Army Research Office
P.O. Box 12211
Res. Tri. Pk., NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARO 34344.1-RT-RIP

11. SUPPLEMENTARY NOTES
Distribution of this document is unlimited.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

As a result of a DARPA DURIP grant in 1995 and an associated NSF Instrumentation award, the University of Rochester has recently acquired two automated vehicles and various computer control and perception hardware. Research supported by these vehicles, commencing now and extending over the next several years, includes selective perception, hierarchical control, cooperation between intelligent agents, and navigation; the applications are cooperative mobile surveillance and monitoring. The mobile cooperating robotics domain has led to several student papers, and research is starting to flow from the new vehicles already.

19960522 089

14. SUBJECT TERMS
low-cost autonomous vehicles; semi-autonomous control; computer vision

15. NUMBER OF PAGES
19

16. PRICE CODE
no charge

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C -	Contract	**PR** -	Project
G -	Grant	**TA** -	Task
PE -	Program Element	**WU** -	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. ***(If known)***

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Vehicles for Selective Vision and Control
Final Report, April 1996 *

Christopher Brown
Department of Computer Science
University of Rochester
Rochester NY 14727-0226
716-275-7852 (vox)
716-461-2018 (fax)
brown@cs.rochester.edu

April 24, 1996

Contents

1 Overview

Thanks to a DARPA DURIP grant and an associated NSF Instrumentation award, U. Rochester has recently acquired two automated vehicles (delivered last week) and various computer control and perception hardware (some delivered, some not). The research goals are selective perception, cooperation, and navigation; the applications are cooperative mobile surveillance and monitoring. The mobile cooperating robotics domain has led to several student papers, and research is starting to flow already ([20, 1]).

2 Cooperating Robots

In future years, the cooperating wheelchairs will be used to investigate issues of cooperation for tasks such as surveillance and monitoring as well as in navigation and material handling.

*This work was supported by DARPA DURIP Grant DAAH04-95-1-0050

To date, work in cooperative robotics has been divided between those who believe robot control is best achieved through symbolic means, including explicit world representation and logical reasoning [42], and those who believe it is best achieved through reactive means, in which robots rely on simple behaviors and intelligence emerges naturally from the interactions among those behaviors [5].

Reactive approaches [36, 49, 12, 40, 15] tend to view cooperating agents as decentralized groups of peers; each agent follows its own reactive programming, and the intelligence in the system supposedly emerges from the interactions among the individual agents. Reactive systems are desirable in that they are robust (a small number of malfunctioning components has little effect on the performance of the system as a whole) and modular (theoretically, the programmer need only think in terms of a single robot; the group behavior will emerge automatically from well-written individual rulebases) [40, 37, 12, 49]. Groups of cooperating reactive agents are often called swarms [36, 49, 12], after the insect societies on which they are modeled.

Symbolic approaches to cooperation [16, 14, 13, 17] use centralized or heirarchical structures, in which some agents are guiding others in their quest for a solution. They use logical planning and explicit world representation to find a near-optimal plan for their subservient agents.

Both methods have their drawbacks. While reactive systems are both robust and (at a single-agent level) understandable, they are also generally inefficient and (at the global level) extremely complex – particularly when complex global behavior is desired, as is the case in many multi-agent systems. Often, reactive systems seem to attain correct global behavior through a combination of luck and sheer persistence on the part of the programmer. When reactive rulebases grow large, reasoning out the varied and complex interactions among rules becomes a difficult process.

Symbolic systems, which generally perform more predictably than reactive systems, and whose global behavior is easier to understand at a glance than is that of an equivalent reactive system, still have a number of nagging problems. They don't deal well with malfunctions; generally, the loss of a component part in a multi-agent planning system leads to the failure of the system – particularly when the malfunctioning component is the system's central arbiter. In addition, difficult tasks can lead to poor behavior on the part of symbolic systems: the combinatorial explosion in forward and backward chaining systems is even more of a problem for multi-agent planners, which typically suffer the additional step of assigning subtasks to component agents. Worse, conflicting subgoals are more of a problem in multi-agent systems than in single-agent planners, because interactions between the individual agents can easily lead to deadlock or goal clobbering [16, 40].

The two most important axes along which cooperative systems differ are *communication* and *organization.* By communication, we mean the method the robots use to exchange information; by organization, we mean the top-down or bottom-up structure of the system [3, 21]. We refer to bottom-up systems, where global behavior emerges from the interactions of many individual autonomous units, as *local.* Top-down systems, where global behavior is designed by the programmer and rigidly enforced by the overall structure of the system, are refered to as *global.*

Figure 1 shows where the areas of study in cooperating robots fall with respect to the categorizations.

2.1 Behavioral cooperation

A number of researchers are investigating cooperation based on behavioral robotics (as proposed by Brooks [5]). Brooks advocates dividing control into several conceptually simple behaviors, all of which run in parallel. Each behavior receives its input straight from the sensors, and each has access to the robot's actuators. The programmer provides a structure whereby behaviors can inhibit each others' outputs (or supress their inputs), thus establishing a means of arbitrating conflicts between behaviors.

The behavioral approach has attracted several researchers in cooperative robotics. One such approach is taken by Mataric [39], who views a society of agents as a single entity. Behaviors are divided across the members of the society. Some reduncancy is, of course, necessary – for example, every agent

	Local Structure	Global Structure
Explicit Communication	Autonomous Cooperating Robots	Heirarchical Systems
Implicit Communication	Observational systems	n/a
No-or-rule-based Communication	Alife, Reactive and Behavioral Systems	Assembly-Line and Manufacturing Robots

Figure 1: Characterization of research into cooperative robotics

has an obstacle avoidance behavior.

Mataric's research develops simple group behaviors that, she claims, can be used as the building blocks for more complex group functions. As with reactive techniques, Mataric's work focuses on local interactions between agents and their environment; it is therefore similar to research in artificial life (Alife) and swarm robotics [10, 11, 49, 12, 36]. Mataric herself mentions the similarity [40], but goes on to differentiate her behavior-based approach from current research in Alife: "However, work in Alife does not typically deal with agents situated in physically realistic worlds. Additionally, it [Alife] usually treats much larger population sizes than the work presented here. Finally, it most commonly employs genetic techniques for evolving the agents' comparitively simple control systems." Mataric's work, in contrast, deals with real robots in the real world, confines itself to smaller groups of agents, and uses reinforcement learning to guide the development of group behaviors.

In her thesis [38], Mataric characterizes AI research as lying in the 2D-space with axes representing cognitive and environmental complexity (see figure 2). Traditional AI research, she claims, deals with complex agents in simple environments. Behavioral and reactive approaches deal with simple agents in complex environments. Mataric attempts to increase both the environmental and cognitive complexity of behavioral systems by:

- allowing multiple agents to act on the world, and
- allowing agents to learn in complex group behaviors.

2.2 Communicating autonomous systems

After a heirarchical organization, the most intuitive approach to multi-agent robotics is to collect a group of single agents and endow each with the ability to communicate in some way with its peers. Here, as with non-communicating reactive (and behavioral) systems, all agents are considered equal. Depending on the environmental context, however, any agent may initiate contact with any other (though generally agents in such systems broadcast their messages instead of sending them point-to-point).

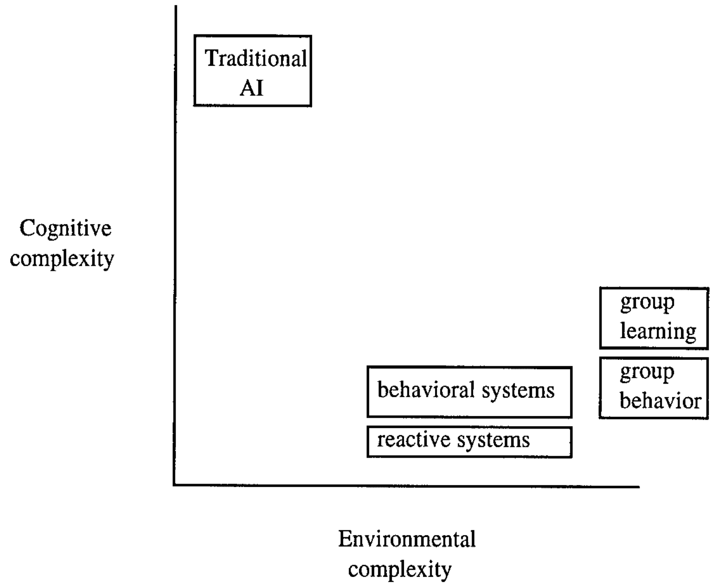


Figure 2: Mataric's characterization of robotics research

Such systems are naturally divided by their application domains. In domains where agents are assumed to be working together toward a common goal, work has focused on the amount and type of communication required to improve performance [2, 44]. In domains where agents are assumed to be self-interested and possibly hostile, research focuses on designing systems where agents can arrange to form optimal-value coalitions [28, 53, 52].

Arkin [2], Parker [44] and Balch [4] attempt to answer the question of whether communication is useful in the shared-goal environment.

Arkin ([2]) examines a simple forage task under varying levels of communication. Since, in his simulation, multiple robots can cooperatively carry an object toward their goal much faster than a single robot can, there is a distinct advantage to cooperating.

With no communicating, Arkin's robots simply wander until they find an object, then pick it up and bring it back to their goals. When the robots can communicate, however, a robot that finds an object broadcasts its location to other robots, who then attempt to assist it in carrying the object back to its goal. Simulations found a moderate decrease in distance travelled per robot in the cooperative scenario, and a large decrease in the number of steps needed to carry an object toward home. Arkin argues that, because the distance travelled is an average over the number of robots, the fact that steps-to-goal is decreasing is an indication that robots are doing more useful work.

In [44], L. E. Parker examines the effect of varying levels of communication on a task where four mobile robots are required to navigate while remaining in formation. She runs simulations under four different communication scenarios: local control only; local control augmented by a global goal; local control augmented by a global goal and partial global information; and local control augmented by a global goal and more complete global information.

In the local-control-only case, each robot uses only its sensor input to determine what to do next. In this case, it is quite possible for robots to become confused. With complete global information, the system is at its best. Agents turn to the right simultaneously as followers predict the plans of the leader.

In contrast, Balch [4] describes a situation where communication is not necessarily helpful to co-operating agents. He investigates three multi-agent tasks: *foraging*, in which agents search for and retrieve goal objects in an arena, *consuming*, in which agents find goal objects and operate on them in the place where they are found, and *grazing*, in which agents' paths must completely cover the space of the arena. Foraging is similar to garbage collection; grazing is similar to a repair task, and grazing is similar to lawn-mowing or floor cleaning.

Balch experiments with three levels of communication: *no* communication, *state* communication and *goal* communication. State communication refers to the case when agents broadcast their internal state. For example, agents broadcasting the fact that they are wandering aimlessly is state communication. Goal communication is the case where agents broadcast information related to a goal – for example, the location of a goal object in a forage task.

Balch finds that the type of task to be performed greatly affects the performance of the various communication schemes. Under both forage and consume, state and goal communication are an improvement over no communication. However, goal communication proves better under the forage task, while state communication proves better under the consume task. (Balch claims that this result is most likely an anomaly.)

In general, then, it seems that cooperating agents experience more success when they have more information. However, Balch's grazing results show that such information can sometimes be gleaned from local environmental conditions without any explicit communication at all. Even for tasks in which environmental information unavailable, more communication does not always equate to more information.

2.3 Communication among Agents with Individual Goals

In environments with multiple agents pursuing individual goals, coalitions are cliques of agents that agree to work together for mutual benefit, possibly to the detriment of the community as a whole [28]. Most work involving coalitions has attempted to determine the best ways to partition agents in order to maximize utility.

Work in this area is heavily influenced by game theory [18, 28, 29, 50, 47, 48], and assumes agents receive "monetary" rewards for achieving tasks, which is maybe not so far-fetched in our networked world but less workable with real robots. The theory here is that agents will join coalitions only if doing so is profitable to them.

Communicating autonomous robots have several desirable features: they are locally organized, which is nice because it reduces the complexity of designing a system, and lends them a robustness that heirarchical systems can not match (though fully reactive and behavioral systems are still more robust). In most domains, communication has been shown to be a useful technique for improving group performance.

However, communicating systems suffer from a number of problems. Most importantly, communicating systems must be composed of groups of homogeneous agents, or at least of groups of agents all of whom use the same method of communication. When such an agent encounters someone from outside its group, it will find it difficult or impossible to cooperate. For example, one of the agents described by Doty would have difficulty to cooperate with a human being, or with one of Balch's agents.

Furthermore, while communication is a useful shorthand between communicating homogeneous agents, it does not relieve agents of the necessity of modeling each other [24]. In communicating societies, the burden of modeling is on the message sender – agents must know that the message they are sending will be accepted by its recipient. In most communicating societies, such modeling is inherent in the structure of the system of communication, and therefore the task of creating the models falls again on the programmer.

Balch's grazing results uncover an interesting direction for research in cooperating robotics. If it is possible for agents to infer information about each other by observing the state of the world, then explicit communication may not be necessary. Systems using implicit communication to gain information about other agents will hereafter be referred to as *observational* systems.

2.4 Future Work

We have discussed cooperative systems with local and global structure that either use explicit communication or do not communicate at all. By *local structure*, we mean systems where the overall societal structure arises bottom-up out of interactions between locally motivated agents. By *global structure*, we mean systems where the overall societal structure is enforced top-down by some sort of heirarchical controller. *Explicit communication* is used by both locally and globally structured systems; it refers to the case where agents intentionally signal each other to communicate parts of their internal state. *Implicit communication* refers to any exchange of information in which the sending agent does not intend to reveal its state; knowledge of other agents is gained by observation of the state of the world.

Heirarchical systems, which are mostly extensions to single-agent planning systems, are globally structured and communicate explicitly. They are formally attractive, and existing research in single-agent planning provides a solid base of knowledge on which to build. Because they model the world and carefully consider their actions, they are able to solve many problems that simpler systems cannot.

However, the reliance of heirarchical systems on the single-agent planning literature lead to multi-agent systems that exhibit many of the same problems that are inherent in planning systems in general [5]. In addition, due to partially to the high complexity of a world that includes many active agents, the multi-agent case of the planning problem has representational, combinatorial and implementational issues that have yet to be fully addressed [39]. Finally, heirarchical systems generally are not robust enough to handle malfunctions in single component agents.

Reactive and *behavioral* systems (those with local structure and no communication) solve many of these problems by synthesizing social organization from the bottom up. In a reactive cooperating system, the loss of an individual agent means little; unless a large percentage of the entire population malfunctions, the system will perform more-or-less as desired. Reactive systems also find it relatively easy to adjust to sudden changes in the world at large.

Reactive cooperative systems have a number of nagging problems. First, because the agents do not model each other or the world, their performance is often haphazard; agents undertake so-called cooperative actions without any sense of whether they are actually helping the agents with whom they are supposed to be cooperating. In addition, the emergent nature of reactive intelligence puts the burden of deciding how to react to various states of the world on the programmer, instead of leaving it with the agents themselves. Because the number of world-states grows exponentially with the number of agents involved, the job of programming reactive systems can easily exceed the abilities of a human programmer. Complex tasks are generally beyond the reach of reactive systems.

Communicating autonomous systems (those with local structure and explicit communication) have been investigated in some problem domains. As implemented, communicating systems have not had much success – their domains are limited, and their performance often lackluster. Many of the problems of reactive approaches plague communicating autonomous systems, too: designers must take into account far too many possible situations for any human programmer to handle comprehensively. Current research focuses on issues of whether to communicate, how much to communicate, and what to communicate. Each of these questions are objects of some controversy.

In our opinion, the ideal cooperative system will combine characteristics from each of these types of systems. It will be retain the robustness of locally organized systems, place the burden of computating responses to world-states on the agent instead of the programmer, model other agents in order to cooperate effectively, and be able to handle complex problems in the real world.

Balch's research [4] motivates a promising area of research into cooperative robotics, one in which locally organized agents communicate implicitly by observing the world-state. These systems fit into the final area of the characterization of cooperative robotics research. We refer to such systems as *observational* systems, and they fulfill many of the specifications for cooperative robotic agents: agents decisions are local, they model the world and each other to arrive at sensible choices of action, they are able to work together efficiently by intuiting each others' internal states.

3 Tracking Known 3-D Objects

Our work with the vehicles will need real-time visual routines, especially tracking and possibly optic flow approximations for navigation. Work has begun on tracking: our idea is to engineer easily-trackable points (with targets or lights) on the lead vehicle. Rodrigo Carcerone has implemented several predictive filters and is comparing their performance. A filter that combines aspects of the lattice filter with recurrent neural nets is promising. This work is combined with a simulator that mimics our digitizer, and which can use either real or graphics-generated imagery. The initial goal is to track "blobs" reliably: once they are tracked, we can use them to recover 3-D state information about the vehicle (its location and orientation) useful for predicting its future state and driving local controllers on the following vehicle.

As part of the research to extract vehicle state from image input, we modified an algorithm originally proposed by David Lowe for use in tracking objects of known geometry to remove certain simplifying assumptions. Experimental results show significant differences in the three versions of the algorithm.

In several research projects at the University of Rochester, a significant subgoal or starting point of further work is the ability to track a set of points in a moving image. Often the geometrical characteristics of these points are know (they occupy known positions on a rigid known shape, for instance). Pioneering work by Gennery [23] and Lowe [35, 34, 33] addresses this basic problem in a projective framework. Recent work in real-time image analysis [8] often makes a simplifying assumption that affine imaging geometry is an adequate model. We are attracted to Lowe's algorithm because of its elegant simplicity, and below we present the algorithm as it appears in the literature and then identify a simplifying assumption that may cause inaccuracies and convergence problems in certain situations. We present two straightforward reformulations that deal with this infelicity, and apply Lowe's technique to them.

David Lowe [35, 34, 33] describes a method for viewpoint and model parameter computation from a known 3-D object, projective imaging assumptions, and the resulting image. The method thus can be used to identify the relative position (translation and orientation) between the camera coordinate system and a local coordinate system on the object, and it can be extended to discovering other parameters, for instance shape parameters of non-rigid objects. He bases his algorithm the application of Newton's method, which assumes that the function relating image appearance and object parameters is locally linear. In general the imaging equations are nonlinear, and so successful application of Newton's method requires starting with an appropriate initial choice for the unknown parameters and still faces the risk of converging to a false local minimum. Possible solutions for the problem of a convergence to a false local minimum are discussed in [35]. For the computation of the Jacobian matrix, Lowe proposes [34] a reparameterization of the projection equations, to simplify the calculation of the necessary derivatives. According to [34] this allows an efficient solution not only of the basic rigid-body problem, but also allows the solution to extend to variable model parameters.

The equations used by Lowe [34] to describe the projection of a three-dimensional model point $\mathbf{p}$ into a two-dimensional image point (u, v) are:

$$(x, y, z) = \mathbf{R}(\mathbf{p} - \mathbf{t}), \quad (u, v) = (\frac{fx}{z}, \frac{fy}{z}) \tag{1}$$

where $\mathbf{t}$ is a 3-D translation vector and $\mathbf{R}$ is a rotation matrix which transforms $\mathbf{p}$ in the original model coordinates into a point (x, y, z) in camera-centered coordinates. These are combined in the second equation with the focal length f to perform perspective projection into an image point (u, v).

The problem is to solve for $\mathbf{t}$,$\mathbf{R}$, and possibly f, given a number of model points and their corresponding locations in an image. In order to apply Newton's method, we must be able to calculate the partial derivatives of u and v with respect to each of the unknown parameters. However, it is not clear at this point how to calculate these partial derivatives for this form of the projection equation. In particular, this formulation does not describe how to represent the rotation $\mathbf{R}$ in terms of its three underlying parameters.

In order to facilitate the calculation of the partial derivatives with respect to the translation parameters, Lowe proposes [34, 33] first to reparameterize the projection equations to *express the translations in terms of the camera coordinate system rather than model coordinates.* The proposed reparameterization is described by the following equations:

$$\begin{aligned}(x, y, z) &= \mathbf{Rp} \\ (u, v) &= (\frac{fx}{z + D_z} + D_x, \frac{fy}{z + D_z} + D_y)\end{aligned} \tag{2}$$

According to Lowe "the variables $\mathbf{R}$ and f remain the same as in the previous transform, but vector $\mathbf{t}$ has been replaced by the parameters D_x, D_y and D_z". The two transforms are equivalent when

$$\mathbf{t} = \mathbf{R}^{-1}\left[-\frac{D_x(z + D_z)}{f}, -\frac{D_y(z + D_z)}{f}, -D_z\right]^T. \tag{3}$$

According to Lowe [34], "in the new parameterization, D_x and D_y simply specify the location of the object on the image plane and D_z specifies the distance of the object from the camera". To compute the partial derivatives with respect to the rotation angles (ϕ_x, ϕ_y, ϕ_z are the rotation angles about x, y and z, respectively), it is necessary to calculate the partial derivatives of x, y and z with respect to these angles.

We believe that Lowe's algorithm embodies a restrictive assumption that can relatively easily be weakened with a resulting increase in the convergence and accuracy properties of the resulting solution.

Suppose the translation vector $\mathbf{t}$ is

$$\mathbf{t} = [t_x, t_y, t_z]^T, \tag{4}$$

the rotation matrix $\mathbf{R}$ is

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{5}$$

and the coordinate vector p of the points in the object coordinate frame is

$$\mathbf{p} = [p_1, p_2, p_3]^T. \tag{6}$$

then using the model described in Eq. 1 the new parameters D_x, D_y, D_z are given by

$$\begin{aligned} D_z &= -(r_{31}t_x + r_{32}t_y + r_{33}t_z) \\ D_y &= -\frac{r_{21}t_x + r_{22}t_y + r_{23}t_z}{(r_{31}p_1 + r_{32}p_2 + r_{33}p_3) + D_z} \\ D_x &= -\frac{r_{11}t_x + r_{12}t_y + r_{13}t_z}{(r_{31}p_1 + r_{32}p_2 + r_{33}p_3) + D_z}. \end{aligned} \tag{7}$$

As can be seen from these expressions, D_z is in fact dependent only on the object pose parameters. On the other hand D_x and D_y are also a function of each point coordinates on the object coordinate frame. It is therefore unacceptable that we try to find a single value for D_x and D_y. In the general case both these parameters will depend on each point. They are not constants — they are only the same for those points for which $r_{31}p_1 + r_{32}p_2 + r_{33}p_3$ has the same value. Therefore we can not use D_x and D_y as defined in Eq. 2. The assumption that is implicit in Lowe's algorithm as published is that the corrections needed for translation are much larger than those due to rotation of the object.

Recent research at Rochester [1] has implemented two solutions to this problem and compared the results with Lowe's original work and with a different implementation by Japanese workers. The results strongly suggest that the new algorithms will be better-suited for accurate real-time solutions.

4 Selective Real-Time Computer Vision

This year we have made progress on selective perception for real-time vehicle control. We are interested in computer vision to support cooperative agents. We allow the agents to be engineered to make some vision problems easier, and to communicate explicitly (we say: to exercise *deictic control*) to ease the technical problems of plan recognition and control. Our ideas are to be instantiated on mobile robots, in particular two automated wheelchairs and possibly other small robots, acquired with ARPA DURIP and NSF Instrumentation Grant funds. We have some experience with small mobile robots [45, 19, 7, 46].

In our domain, autonomous or human-controlled vehicles interact through sensing and (minimal) communication. The insertion of humans and symbolic communication in the loop is, we believe, both realistic and interesting. Many difficult vision problems remain but they occur in constrained contexts so that certain high-level problems, such as "segmenting" out the relevant signal and choosing the next relevant action, disappear. The result is that vision can be robustly applied to a constrained problem and that the computer can be used effectively to enhance human capabilities rather than trying immediately to replace all human capabilities.

Fig. 3 shows the visual inputs and behavioral outputs for our automated vehicle. *Deictic* inputs are explicit signals (like a turn signal) that indicate a vehicle's intentions or communicate warnings, hints, commands, or parameters. *Implicit* inputs can be *engineered* (e.g. lights or targets in known geometrical configurations, known shapes) or *natural* (e.g. flow fields from the landscape or unknown obstacles).

We are doing highly selective vision for tracking known targets (*engineered tracking* as opposed to *natural tracking* for arbitrary objects), with the goal of recovering (observing and estimating) the state of a companion vehicle. *Implicit* information is extracted from a target as a result of known physics and geometry: *Explicit* signals communicate arbitrary messages by convention. We to obtain both implicit three-dimensional information (location, orientation, and their derivatives) and explicit signals from monocular images.

The foundational capability we have been addressing is the problem of following another vehicle. By tracking a set of points (some number greater than four) that are engineered to be simply related to the local coordinate system of the other vehicle, its location and orientation in space may easily be determined. This information gives us a state 6-vector (locations and Euler angles, say) at an instant k:

$$\mathbf{x}(k) = \begin{bmatrix} X \\ Y \\ Z \\ O \\ A \\ T \end{bmatrix} (k).$$

Differencing this vector gives an approximation to relevant velocities and accelerations in space and

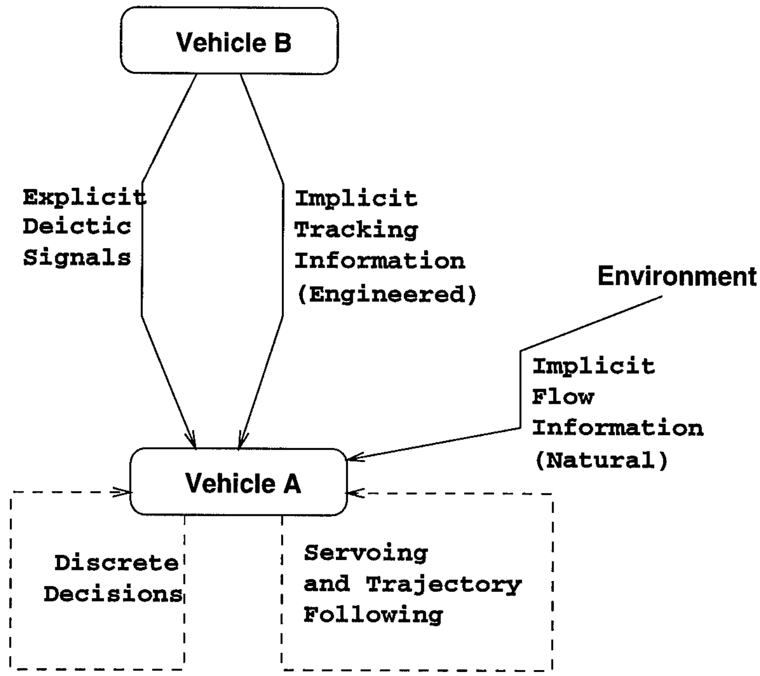


Figure 3: Visual Inputs and Control Outputs for the Robot Vehicle

orientation, which can be used along with a dynamic model of the following vehicle to determine the inputs to its accelerator, brake, and steering.

We have developed techniques for extractind reliable information from visual tracking of engineered, implicit data. Related necessary work is to produce accurate models of the vehicles involved, and to produce effective control (ultimately, effective behavior). We assume that the control we have over a vehicle is to set its steering rate and acceleration; first order and second order control, respectively.

Tracking of simple target features is a foundational capability for any aspect of real-time vision [6]. At Rochester we have several successful trackers for binary and grayscale features. Tracking known 3-D geometrical shapes has a long history and is still an active topic today [22, 51, 32, 25, 43, 41, 27]. Generally, the model-based vision and known-object tracking problems have been phrased in terms of complex minimization problems over large parameter spaces of models and their geometric properties. In contrast, we use a simple affine location-finding algorithm. Tracking features that are assumed to lie in a 3-D affine frame of reference has become an important technique since it simplifies calculations with little loss of accuracy in many practical situations [9, 31, 26, 30].

Once our hardware arrives, we plan to mount a set of easily-trackable targets (lights) in a known configuration on one or both of the vehicles. After vehicle A acquires and identifies vehicle B's lights, tracking them will determine all B's six locational and orientational degrees of freedom (its (X, Y, Z) or (range and direction), and its orientation in A's coordinates).

Vision processing converts the target images to image-coordinate points. Assume we always see all points and always know which image point corresponds to which target point. The simplest workable target is four lights: one at the origin and one each at unit distance in the vehicle's right-handed X, Y, Z local coordinate system. Call these target points O, X, Y, Z. In fact we plan to use a cube of eight lights, which allows four measurements in each of the three coordinate directions, thus increasing accuracy.

The direction of the vehicle is easily determined from estimating the direction of point O, for instance, and using its known location in vehicle coordinates. Distance can be calculated from the image distances between target points of known physical distance and the laws of perspective. Last, we want to infer the orientation of the target coordinate system relative to the camera coordinate system from a single image. Approximate the projective transform as affine, and ignore scaling. (These assumptions would all be exactly true for orthographic projection).

Let "camera coordinates" C be a 3-D system whose Z direction is out along the line of sight and "image coordinates" be a 2-D system making up the $X-Y$ plane of camera coordinates. Move the image of O to the origin of image (hence camera) coordinates. This removes a translation term irrelevant to the problem of figuring orientation. Now the entire camera transformation is just a rotation of a 3-D point followed by projecting away the Z-element of the result to get a 2-D image point.

Represent a coordinate system by three column vectors giving the directions of the X, Y, Z axes. One way to do this is just to write down the X, Y, Z unit vectors. The resulting matrix is orthonormal.

Let the L (LAB) coordinate system be the one in which we express all points:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The columns of a 3×3 coordinate system C matrix represent three 3-D points, but also the rotation transformation needed to map the base L coordinate system (now considered as three points) into C (itself now considered as points) since

$$C = CL.$$

Let the target coordinates be L and express the camera coordinates C in terms of them. The camera acts on points in the world, expressed in L, yielding J (the image). The physical camera transform K maps 3-D world points X to 2-D image points:

$$J = KX.$$

Here K is a 2×3 camera transform matrix, X a $3 \times N$ matrix of (X, Y, Z) points, and J is a $2 \times N$ matrix of (X, Y) image-coordinate points.

But if $X = L$, the identity matrix (i.e. if we "take a picture" of the lab coordinate system,) then

$$J = K.$$

Since C is orthonormal, its third row is the cross product of the first two rows, and the image J itself gives us the first two rows. This observation lets us calculate C, which is the camera in terms of L, and $C^T = C^{-1}$ is the "laboratory" (target) coordinate system in terms of C, which is the orientation of the lead vehicle in terms of the observing vehicle's camera coordinates.

We have started a simulation study that duplicates the projection and digitization processes in on-board cameras, blob tracking, geometric interpretation and state estimation, and vehicle guidance (Fig. 4).

In some versions of our implementation, the correspondence problem is mostly solved by rotating the cube so that none of the lights is at the same height (Y coordinate). (A good transformation of the "unit" cube is first a Z rotation by .463 radians, then an X rotation by .221 radians.) This means that lights are unambiguously identified by their relative height if the vehicles are on a plane. We also attack the correspondence problem with predictive filters. We are using the simulator to study the behavior of various predictive filters.

5 The Hardware and Software Components

Major hardware components and associated software are the following. All the following but the Canon heads are ordered, all but the Canon and the RWII computers are in house and being integrated currently.

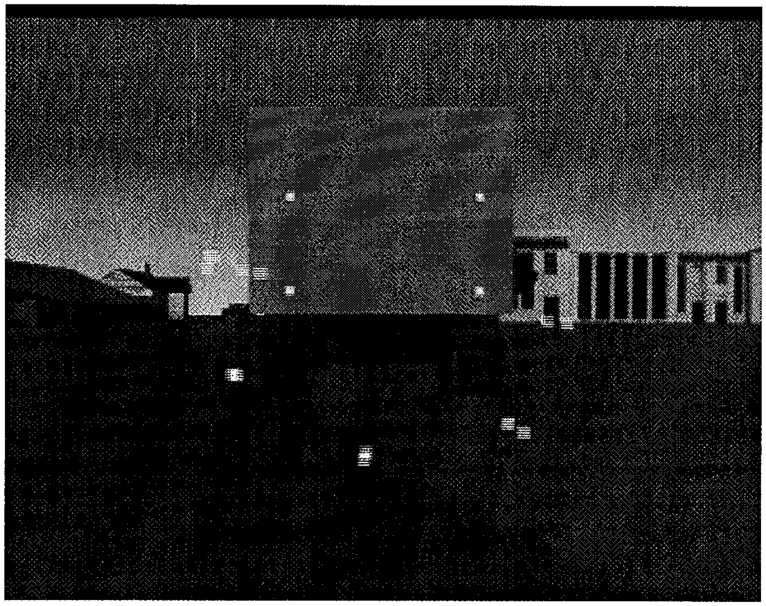


Figure 4: Frame from graphic version of the following problem. Four lights are mounted on the lead truck, and some moving noise points are visible.

- Two computer-controllable, motorized wheelchairs from KIPR, with microcontrollers running ARC.
- Two twin-pentium-based control computers from RWII, running Linux.
- Two Matrox meteor digitizers, with Linux drivers from RWII.
- Two FM remote TV setups so we can monitor or even digitize the television from the vehicles
- Two Wireless Ethernets
- Sundry cameras.
- We are considering two Canon pan-tilt heads.

In recent work to analyze the vehicle's motors preparatory to developing simulations and controllers, Roger Gans of Mechanical Engineering at the UR has analyzed performance curves sent by the manufacturers. His report is as follows.

We have acquired two wheel chairs driven by two independent identical motors. The manufacturer has provided us with torque, speed and current data at 24 volt excitation (see Fig. 5), from which we have deduced the nature of the motors. The power dissipation by friction in the motors seems to be proportional to the motor speed, from which the frictional torque is found to be approximately constant at 0.141 Nm, consistent with the single datum shown in Fig. 5. Finally the net available mechanical torque T can be written in terms of the armature resistance R and the rotation rate n (in Hz) as

$$T = \frac{(0.6(V - 0.6n)}{(2\pi R)} - 0.141.$$

Control is possible either by controlling the voltage V or the resistance R. Fig. 6 shows the torque as a function of V and n, and Fig. 7 as a function of R and n. The test data were apparently generated at an armature resistance of 0.172 O.

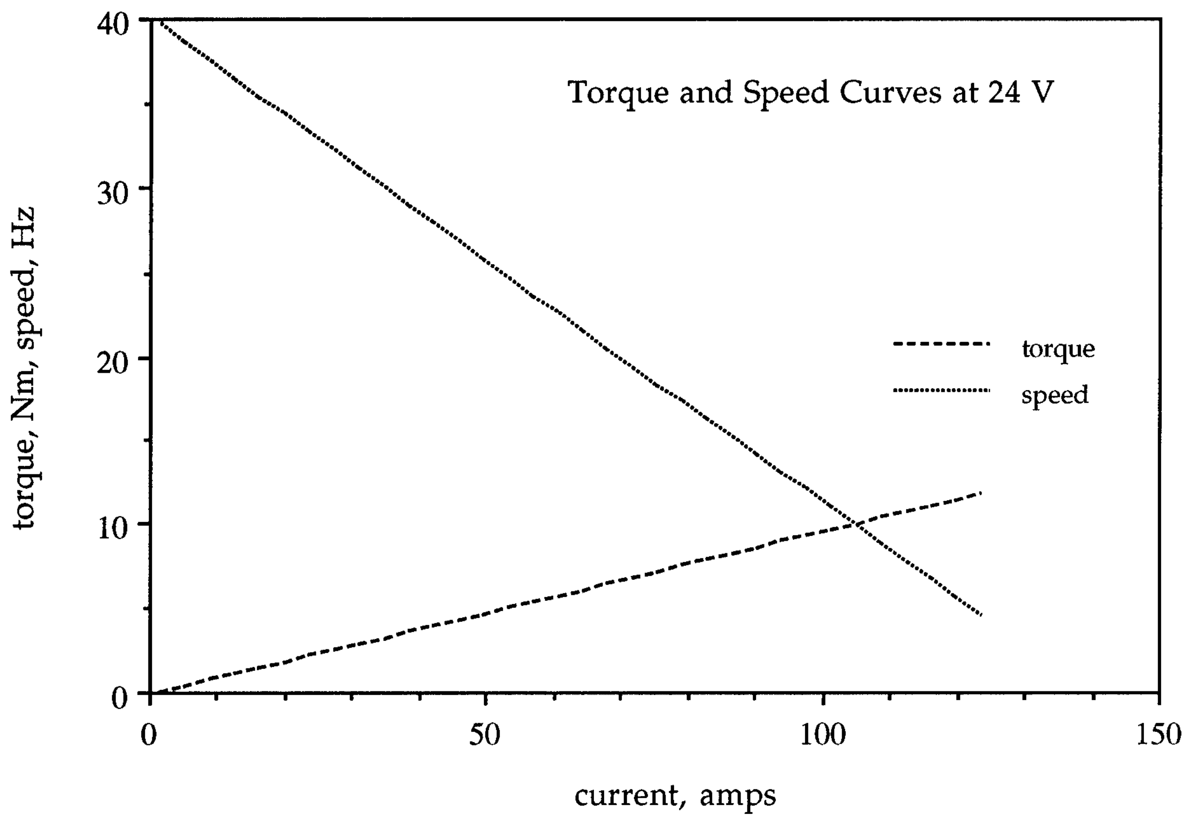
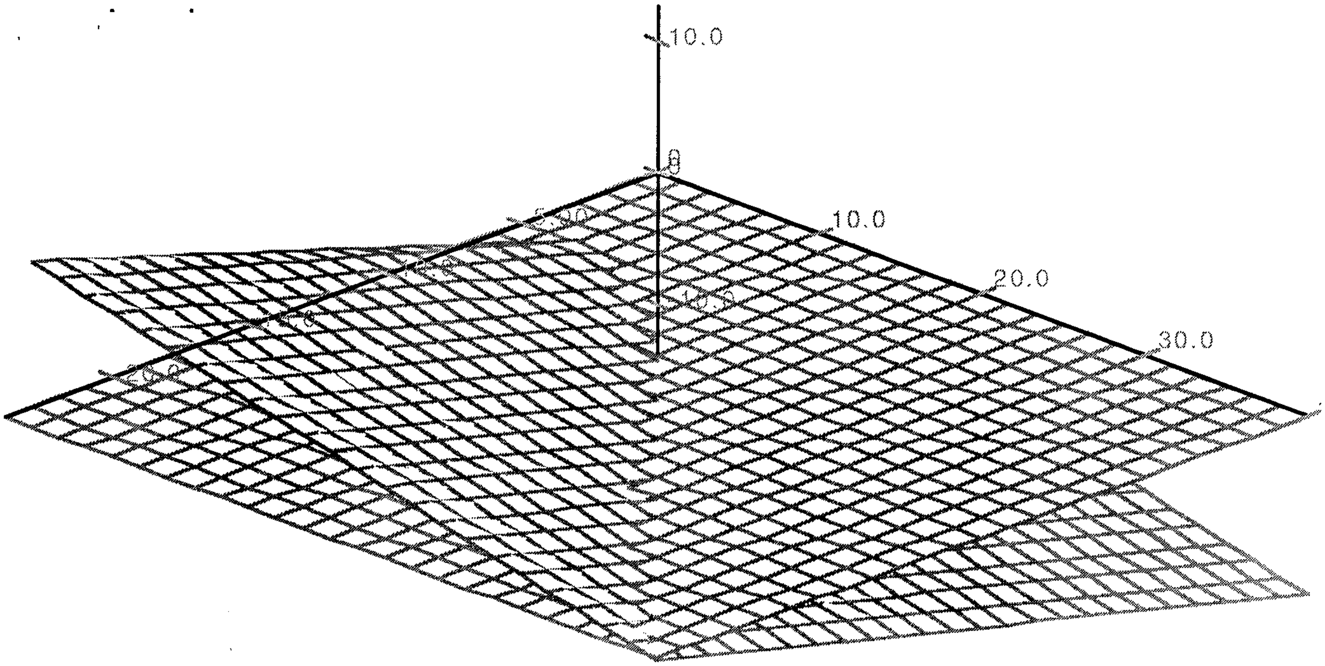
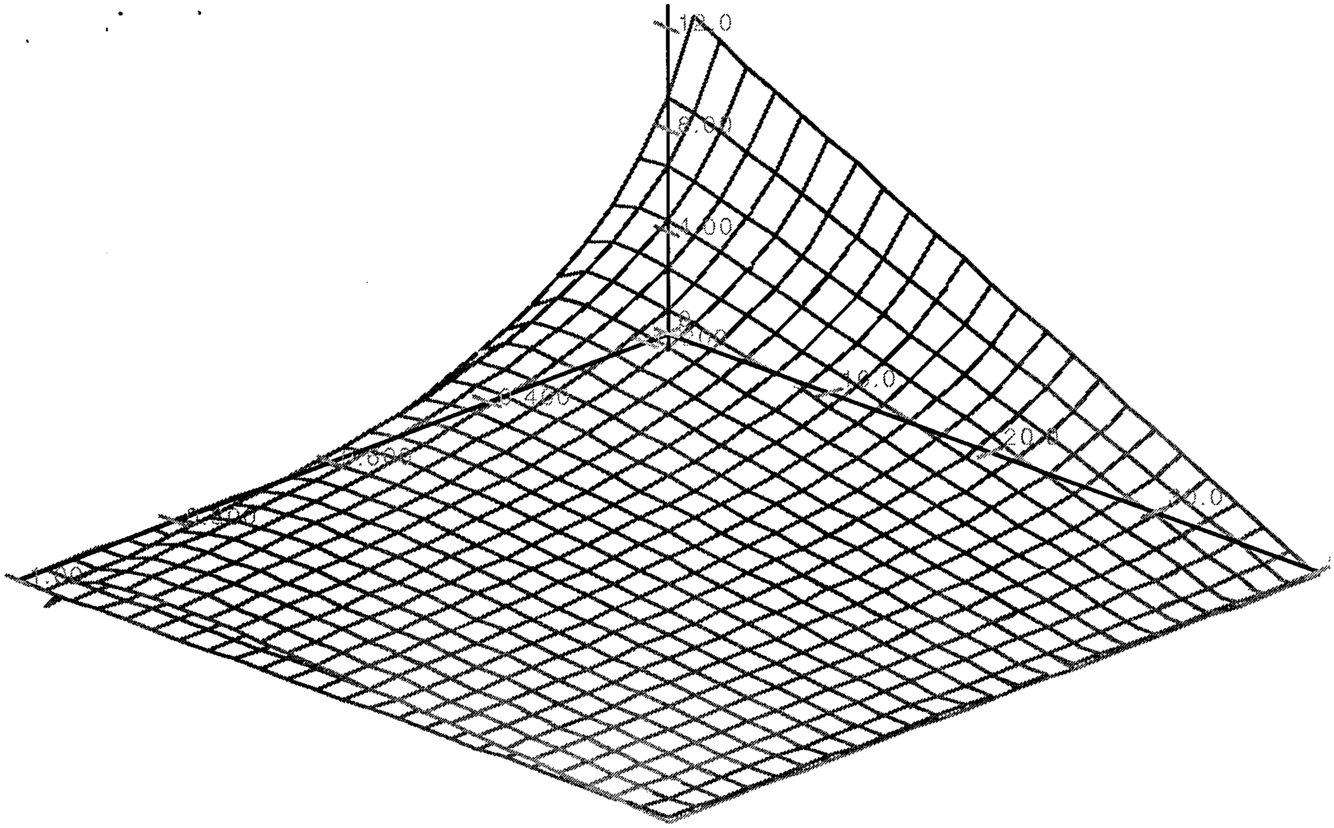

Torque and Speed Curves at 24 V
torque
speed
torque, Nm, speed, Hz
current, amps
0
10
20
30
40
0
50
100
150

Figure 5.



Torque vs Voltage and Rotation Rate at Nominal Resistance = 0.172Ω

Figure 6.



Torque vs Resistance and Rotation Rate at V = 24 v

Figure 7.

References

[1] Helder Araujo and Christopher Brown. A note on lowe's tracking algorithm. Technical Report TR 610 (submitted for publication), Computer Science Dept., U. Rochester, April 1996.

[2] R. C. Arkin, T. Balch, and E. Nitz. Communication of behavioral state in muti-agent retrieval tasks. *Proceedings, IEEE International Conference on Robotics and Automation*, 3, 1993.

[3] Hajime Asama. Trends of distributed autonomous robotic systems. pages 3 – 8, 1994.

[4] T. Balch and R. C. Arkin. Communication in reactive multi-agent robotic systems. *Autonomous Robots*, 1(1):27–53, 1994.

[5] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), March 1986.

[6] C. Brown and D. Terzpoulos, editors. *Real-time Computer Vision.* Cambridge University Press, 1994.

[7] C. M. Brown. Mobile robotics 1994. Technical Report 588, Department of Computer Science, University of Rochester, Rochester, NY, June 1995.

[8] C. M. Brown and D. Terzopoulos, editors. *Real-Time Computer Vision.* Cambridge University Press, 1994.

[9] R. Cipolla and N. Hollinghurst. Visual robot guidance from uncalibrated stereo. In C. Brown and D. Terzpoulos, editors, *Real-time Computer Vision*, pages 169–188. Cambridge University Press, 1994.

[10] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed optimization by ant colonies. *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 134 – 142, 1992.

[11] J. L. Deneubourg, G. Theraulax, and R. Beckers. Swarm-made architectures. *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 123 – 133, 1992.

[12] K. L. Doty and R. E. Van Aken. Swarm robot materials handling paradigm for a manufacturing workcell. *Proceedings, IEEE International conference on Robotics And Automation*, 1, 1993.

[13] Edmund H. Durfee and Jeffrey S. Rosenschein. Distributed problem solving and multi-agent systems: Comparisons and examples. *Thirteenth International Distributed Artificial Intelligence Workshop*, pages 94 – 104, 1994.

[14] E. Ephrati, M. Pollack, and J. Rosenschein. A tractable heuristic that maximizes global utility through local plan combination. *First International Conference on Multi-Agent Systems*, 1995.

[15] E. Ephrati, M. Pollack, and S. Ur. Deriving multi-agent coordination through filtering strategies. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, August 1995.

[16] E. Ephrati and J. Rosenschein. Divide and conquer in multi-agent planning. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, July-August 1994.

[17] E. Ephrati and J. S. Rosenschein. Multi-agent planning as a dynamic search for social consensus. *Thirteenth International Joint Conference ON Artificial Intelligence*, pages 423 – 429, 1993.

[18] Eithan Ephrati and Jeffrey S. Rosenschein. The clarke tax as a consensus mechanism among automated agents. *AAAI*, 1:173–178, 1991.

[19] Olac Fuentes, Rajesh P. N. Rao, and Michael Van Wie. Hierarchical learning of reactive behaviors in an autonomous mobile robot using stochastic hillclimbing. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 1995*, Vancouver, B.C., Canada, October 1995. to appear.

[20] Roger F. Gans. Following behavior using moving potential fields. Technical Report TR 603 (submitted for publication), Computer Science Dept., U. Rochester, January 1996.

[21] L. Gasser and M. N. Huhns. Themes in distributed artificial intelligence research. *Distributed Artificial Intelligence*, 2, 1989.

[22] D. Gennery. Tracking known three-dimensional objects. In *Proc. 2nd National Conf. Artif. Intell.*, pages 13–17, 1982.

[23] Donald B. Gennery. Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, 7(3):243–270, 1992.

[24] Piotr J. Gmytrasiewicz, Edmund H. Durfee, and David K. Wehe. The utility of communication in coordinating intelligent agents. *AAAI*, 1:166–172, 1991.

[25] C. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*. MIT Press, 1992.

[26] M. Jagersand and R. Nelson. Adaptive differential visual feedback for uncalibrated hand-eye coordination and motor control. Technical Report 579, Department of Computer Science, University of Rochester, Rochester, NY, December 1994.

[27] M. Jagersand and R. Nelson. Visual space task specification, planning, and control. In *Proceedings, IEEE Symposium on Computer Vision*, 1995.

[28] S. Ketchpel. Forming coalitions in the face of uncertain rewards. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, July-August 1994.

[29] Sarit Kraus and Jonathan Wilkenfeld. The function of time in cooperative negotiations. *AAAI*, 1:179–184, 1991.

[30] K. Kutulakos and M. Jagersand. Exploring objects by purposive invariang-based tangential viewpoint control. In *Proceedings, IEEE Symposium on Computer Vision*, 1995.

[31] K. Kutulakos and J. Vallino. Affine object representations for calibration-free augmented reality. In *Proceedings, IEEE Virtual Reality International Symposium*, 1996.

[32] D. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1991.

[33] David Lowe. Solving for the parameters of object models from image descriptions. In *Proc. ARPA Image Understanding Workshop*, pages 121–127, College Park,MD, Apr. 1980.

[34] David Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, Mar 1987.

[35] David Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[36] G. Lucarini, M. Varioli, R. Cerutti, and G. Sandini. Cellular robotics: Simulation and hw implementation. *Proceedings, IEEE Internation Conference on Robotics and Automation*, 3, 1993.

[37] Maja J. Mataric. Behavior-based systems: Key properties and implications. *Proceedings, IEEE International conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems*, 1992.

[38] Maja J. Mataric. Interaction and intelligent behavior. *Technical Report, MIT*, 1994.

[39] Maja J. Mataric. Designing and understanding adaptive group behavior. *Adaptive Behavior*, pages 51 – 80, 1995.

[40] Maja J. Mataric. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2 – 4):321 – 331, 1995.

[41] D. Murray, I. Reid, K. Bradshaw, P. McLaughlan, P. Sharkey, and S. Fairly. Active exploration of dynamic and static scenes. In C. Brown and D. Terzpoulos, editors, *Real-time Computer Vision*, pages 117–140. Cambridge University Press, 1994.

[42] Nils J. Nilsson. Shakey the robot. Technical Report Technical Note 323, SRI International, April 1984.

[43] N. Papanikolopulos, P. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. *IEEE Transactions on Robotics and Automation*, 9(1):14–35, February 1991.

[44] L. E. Parker. Designing control laws for cooperative agent teams. *Proceedings, IEEE International Conference on Robotics and Automation*, 3, 1993.

[45] Rajesh P. N. Rao and Olac Fuentes. Perceptual homing by an autonomous mobile robot using sparse self-organizing sensory-motor maps. In *Proceedings of the World Congress on Neural Networks 1995*, Washington, D.C., July 1995. to appear.

[46] J. P. Rosca and T. Riopka. A constraint-based control architecture for acting and reasoning in autonomous robots. In *Proceedings, AAAI Spring Symposium: Lessons learned from implemented software architectures for physical agents*, Stanford, March 1995.

[47] T. W. Sandholm and V. R. Lesser. Coalition formation among bounded rational agents. *IJCAI*, 1:662–669, 1995.

[48] T. W. Sandholm and V. R. Lesser. Equilibrium analysis of the possibilities of unenforced exchange in multiagent systems. *IJCAI*, 1:694–701, 1995.

[49] D. J. Stilwell and J. S. Bay. Toward the development of a material transport system using swarms of ant-like robots. *Proceedings, IEEE International Conference on Robotics and Automation*, 1, 1993.

[50] S. Sycara. Multiagent compromise via negotiation. 2:119–137, 1989.

[51] G. Verghese, K. L. Gale, and C. R. Dyer. Real-time motion tracking of three-dimensional objects. In *Proc. Int. Conf. on Robotics and Automation*, pages 1998–2003, 1991.

[52] Gilad Zlotkin and Jeffrey S. Rosenschein. Negotiation and task sharing among autonomous agents in cooperative domains. *Eleventh International Joint Conference on Artificial Intelligence*, pages 912 – 917, 1989.

[53] Gilad Zlotkin and Jeffrey S. Rosenschein. Coalition, cryptography and stability: Mechanisms for coalition formation in task oriented domains. *The National Conference on Artificial Intelligence*, pages 432 – 437, 1994.