1

Final Report
ONR Contract Number N00014-91-J-1577
Principal Investigators: Drew McDermott and Gregory Hager
Yale University Department of Computer Science

1 Summary of Technical Results

The object of the Yale Knowledge-Based Planning project is to find a unified theory of robot planning and sensing. We achieved several results in this direction, including the development of an architecture for a reactive planner, the implementation and testing of a theory of map making by a mobile robot, the development of an algorithm for reasoning about constraints on image hypotheses, and preliminary work on high-speed visual tracking.

Our overall model of planning is this: An agent must be continously executing plans in order to make progress on its goals. These plans are driven by sensors, and can normally cope with deviations from expected results without intervention from the planner. When intervention is required, the planner starts from scratch, generating plans and then revising them on the basis of their projected results. The projector contains a probabilistic model of the world that allows the planner to forecast probable errors.

Our work on sensing has focused on these areas:

1. Theoretical foundations for set-based decision-making algorithms.
2. Visual tracking and vision-based *control of servo systems.*
3. Comparison of set-based and statistically-based estimation.

We will discuss all these areas in more detail.

1.1 Reactive Planning Architecture

"Reactive" plans are simply robot plans that involve explicit steps for using sensors and then reacting to the data gathered. A reactive plan differs from a traditional plan in two principal ways: it must have conditionals that branch based on the outcome of sensory tests; and it must have local variables that get bound to the sensed objects and their properties. The added complexity makes the planning problem harder, and so does the fact that the planner does not know everything about the world. In fact, most work on reactive planning falls into two categories: studies based on robot plans written entirely by hand (so that there is no real planning at all); and studies based on extreme special cases in which the form of the plans is given, and the planner then tunes various parameters.

What we have developed is an architecture, called XFRM, that allows us to go a bit beyond these limits. We allow plans to be written in a flexible and general language, the Reactive Plan Language. It contains a uniform notation for referring to objects discovered by the sensors. It also provides convenient ways for a planner to transform plans. For example, substeps of a plan can be tagged, and the tags then used in ordering statements and policies that constrain how the substeps are to be executed.

One of our main results in this area was the development of an efficient and clean plan projector. When dealing with complex reactive plans, a key reasoning strategy is to mentally simulate a plan and see how well it accomplishes the goals, what bugs it has, and what resources it consumes. This "mental simulator" is called a *plan projector*. Its output is a set of scenarios showing what might occur when the plan is executed. We have developed a simple algorithm that takes rules describing the effects of actions, and builds a timeline summarizing the effects of an entire plan. The rules are stated in a simple predicate-calculus format, like this:

```
(E->P (AND (LOC ROBOT ?WHERE)
           (LOC ?OB ?WHERE)
```

```
          (HAND-INTERFERENCE ?OB ?HAND ?P))
      (PICKUP ?OB ?HAND)
      ?P
  (LOC ?OB ?HAND))
```

which says that a **PICKUP** action succeeds in getting an object into the robot's hand with probability **?P** if **HAND-INTERFERENCE** occurs with success probability ?P. (Other rules can be supplied that detail under what circumstances **HAND-INTERFERENCE** occurs and reduces the probability.) The projector also includes rules that model autonomous Poisson-distributed events. The rule (P->E *P d E*) specifies that over any interval where *P* is true, the expected time to the next occurrence of *E* is *d*. We have developed a formal semantics for these and the other rule types, and shown that the program generates timelines with the probabilities given by the formal semantics.

Empirical tests show that for typical rule sets the projector runs in time proportional to the square of the number of events generated. It achieves this efficiency by a variety of optimizations, including caching the results of retrievals at timepoints. Actually, this strategy is a necessity because a probabilistic test can obviously not be counted on to give the same results when run twice. But the effect is that the system rarely has to sweep back far through the timeline looking for answers to a query; and when it does, the answers are cached for the next occurrence of that query. The projector is so efficient that we can run it several times for a typical plan, yielding a sample of possible scenarios.

We have implemented a robust interface between the planner and the plan-execution module that allows swapping in of a new plan at any time. We have run experiments in our "delivery world" that show that the system is able to achieve significant improvements in performance times simply by planning simultaneously with execution. Typically, in cases where the planner can run fast enough to "beat" the interpreter, the plan it swaps in embodies speedups over the default plan that compensate for the time and side effects incurred while executing the default.

Transformational planners as yet lack a theoretical basis of the sort that underlies refinement planners. A transformation can make an arbitrary change in a plan, and only the projector can verify that the new plan is an improvement. To make the process run as reliably as possible, plans have to be represented in a transparent manner that enables the planner to see the purposes of the pieces of behavior it encapsulates. We have provided declarative constructs for expressing these purposes. In particular, where possible we express sensor tests using a **BELIEF** construct that makes it explicit which beliefs about the agent's environment are being polled. We have also developed a Prolog-like "meta-language" **XFRM-ML** for expressing transformation rules and their associated preconditions.

1.2 Interval-Based Inference on Sensor Data

Many sensor-data-processing tasks can be be phrased in terms of finding values of parameters that satisfy given constraints. For example, determining whether a certain set of edges in an image could be an instance of an object model can be thought of as finding values for the object's parameters that account for the given edges. Our approach to finding such values is to start with intervals containing the correct values, and gradually prune away subintervals that are inconsistent, until we are left with subintervals that are guaranteed to contain at least one point satisfying the constraints.

Over the course of this project, we

- Designed and implemented the basic constraint-satisfaction algorithm.
- Implemented a distributed version of the algorithm. The resulting implementation is significantly faster (i.e. parallel) and more fault-tolerant.
- Implemented a data selection technique that reduced the time needed to solve some benchmark problems to less than 1 second (essentially real-time operation).

- Implemented and tested the algorithms on data requiring descrimination and comparisons among multiple objects or targets.
- Implemented a version of the algorithms for use in unstructured domains. This significantly increases the domain of applicability of the algorithms, making it possible, for example, to apply them outdoors.

Our algorithms for set-based decision-making explictly recover the parameters of geometrical or physical models until decision-specific accuracy criteria have been met. It is possible to supply any number of decision criteria that are evaluated in parallel as parameter recovery is performed. We have been able to show that the algorithm we use for evaluating decision criteria is correct and complete except for a vanishingly small set of problems. Correctness means that only physically correct decisions will be made as long as the input data are consistent with the parametric models supplied to the algorithm. Completeness means that the algorithm will terminate on all inputs. We have shown that the algorithm we use will terminate on all inputs except for certain boundary cases that are typically a set of measure zero in the space of algorithm inputs.

These results hold for an extremely wide variety of problem settings including problems where the number and type of geometric models is not known a priori. This means that our algorithm is effectively a decision procedure even when both segmentation and parameter fitting must be performed simultaneously.

1.3 Map Building

Similar set-based methods have been employed by us for working in mobile robot mapping and navigation. The problem of mobile robot navigation is getting the robot to a place it's been before. There are several issues that must be resolved in systems which build such representations. We must pin down what we mean by 'place,' and do so in a way that supports efficient recording and recognition of places. Several methods for automated map construction have been reported, but they all suffer from the problem of *error accumulation*. Since all sensors have noise, and sensor interpretation often depends on violable assumptions about the real world, any system which attempts to build a consistent representation of its environment will make errors. In particular, the robot's decision that it has been somewhere before (more generally, that two places are the same) can never be perfectly justified and always involves some chance of error. If a mistaken identification is allowed to persist, then attempts to make the rest of the map consistent with it will eventually turn the whole map into garbage. Hence, some mechanism must be provided by which these errors can be detected and corrected. Interestingly, this issue has been largely ignored in the literature, with the primary emphasis going to reducing errors entering the map to begin with.

This problem of autonomous environmental representation (the 'map-learning' problem) has been studied for some time, from a number of viewpoints. There are two basic types of approach—metric and topological. The metric approach attempts to build up a detailed geometric description of the environment from perceptual data, while the topological approach concentrates on learning the qualitative shape of the robot's own paths. We have developed a hybrid model, in which the robot tries to learn the *metric* shape of its paths. There are two fairly obvious reasons for this move: the metric information can help in distingishing between perceptually similar places; and the metric information is useful in deciding where to go and how to get there when the map is used.

So a map includes a graph with nodes representing 'places', i.e., connected regions of a particular type, and arcs labelled with sequences of actions, generally concluding with an approacher. (Presently, we deal only with 'point-like' places, small regions which can be treated as single points; the complexities of shape representation will be investigated in future work.) However, there is more to the graph. Each node has a record of what the place looks like, and what its position is with respect to other nodes. As the robot wanders through the world, it adds new nodes to the graph and refines its estimate of the positions of old nodes.

There are several kinds of error that can occur. Some are relatively simple to correct, including errors in position estimates. The hard ones are errors of identification, where two places are mistakenly assumed equal, or one place is mistakenly broken in two. These errors are dealt with by the following techniques. First, when the robot is not sure of its location, it maintains multiple tracks through the map, until all but one have been disconfirmed by later reports. Second, if two tracks look quite similar, it merges the places along them. Third, it keeps statistics on the position of a place, and if the position has a multimodal distribution, it considers breaking it into two or more places.

Experiments with these techniques on a simulated world show that they usually converge to a correct map, even in the presence of noise. We plan next to try running them with real visual data.

We have run experiments comparing set-based methods with classical statistical estimation methods for map representation. We undertook a study to compare statistical estimation methods with set-based methods for the purpose of robot navigation. We found that set-based methods typically outperform statistical methods when the estimation problem has low-dimension and is nonlinear. As it turns out, many of the estimation problems faced in robot navigation have these properties, so set-based methods would be expected to outperform statistical methods. We have run both simulated and real experiments with a mobile robot system and have verified that this is true.

Final Report
ONR Contract Number N00014-91-J-1577
Principal Investigators: Drew McDermott and Gregory Hager
Yale University Department of Computer Science

2 Publications, Presentations and Reports

- Sami Atiya and Gregory D. Hager (1991) Real-time vision-based robot localization. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 639–643. IEEE Computer Society Press.
- Sami Atiya and Gregory D. Hager (1994) "Real-Time Vision-Based Robot Localization," *IEEE Trans. on Robotics and Automation*, in press
- Michael Beetz, (July 1992), Panelist at the Panel "Planning and Scheduling" (Workshop "Implementing Temporal Reasoning," AAAI-92
- Michael Beetz, (August 1992), Talk on "Improving and Debugging Reactive Plans that Contain Declarative Goals." German Research Center for Artificial Intelligence, Inc. (DFKI).
- Michael Beetz, (August 1992), Talk on "Improving and Debugging Reactive Plans that Contain Declarative Goals." Bavarian Research Center for Knowledge-based Systems (FORWISS), Germany.
- Michael Beetz (August, 1993), "Improving Robot Plans During their Execution," presentation at the Bavarian Center for Knowledge-based Systems/ University of Erlangen: the Technical University of Darmstadt: and the German Research Center for Artificial Intelligence, Saarbruecken;, Germany
- Michael Beetz, M. Lindner, and J. Schneeberger (1992) Temporal Projection for Hierarchical, Partial-order Planning. *Proceedings of ISAI-92*, AAAI Press.
- Michael Beetz and Drew McDermott (1992) Declarative goals in reactive plans. In James Hendler (ed.) , *Proc. First Int. Conf. on AI Planning Systems*, San Mateo: Morgan Kaufmann, pp. 3–12
- Michael Black (1991) "Optic flow and motion discontinuities over long image sequenes: Experimental Results," invited presentation at *IEEE Workshop on Visual Motion*, Princeton, NJ, October
- Michael Black and P. Anandan (1991) Robust dynamic motion estimation over time. *Proc. Computer Vision and Pattern Recognition*, CVPR-91, Maui, Hawaii, , pp. 296–302.
- Michael Black and P. Anandan (1991) Dynamic motion estimation and feature extraction over long image sequences. *Proc. IJCAI Workshop on Dynamic Scene Understanding*, Sydney, Australia, August
- Sean Engelson (1992), "Active Place Recognition Using Image Signatures." SPIE 1992
- Sean Engelson and Niklas Bertani (1992) "ARS MAGNA: The Abstract Robot Simulator Manual." Yale Computer Science Report 928.
- Sean Engelson and Drew McDermott (1991) Image signatures for place recognition and map construction. SPIE Technical Symposium on Advances in Intelligent Robotic Systems.
- Sean Engelson and Drew McDermott (1992) "Active Place Recognition Using Image Signatures", Proceedings of SPIE Sensor Fusion V, November 1992

- Sean Engelson and Drew McDermott (1992). "Maps Considered As Adaptive Planning Resources". AAAI Fall Symposium on Applications of AI to Real-World Autonomous Mobile Robots
- Gregory D. Hager (1991b) Set-based estimation: Towards task-directed sensing. In *Proceedings of Melecon '91*, pages 1205–1209.
- Gregory D. Hager (1991a) "Sensor data fusion and robotics." Invited Lecture for the first Workshop on Mathematical Problems in Robotics.
- Gregory D. Hager (1991c) Using resource-bounded sensing in telerobotics. In *Fifth International Conference on Advanced Robotics*, pages 199–204.
- Gregory D. Hager (1991) Towards geometric decision making in unstructured environments. In *Proc. 1991 International Workshop on Intelligent Robots and Systems*, Bellingham, WA, pp. 1412–1417.
- Gregory D. Hager (1992) "A Constraint-Based View of Selective Perception". Proceedings of the AAAI Spring Symposium on Selective Perception, Stanford, CA, March 1992.
- Gregory D. Hager (1992) "Constraint Solving Methods and Sensor-Based Decision Making" *Proc. IEEE Conf. on Robotics and Automation.*
- Gregory D. Hager. (April 1992). "Sensor Data Fusion," a lecture delivered at Red Stone Arsenal, Huntsville, Alabama.
- Gregory D. Hager. June (1992). "Sensor-Based Decision Making" presented at the DLR (German Space Organization), Oberpfaffenhofen, Germany.
- Gregory D. Hager (1992). "Towards task-directed planning of cooperating sensors," presentation at the SPIE Sensor Fusion Workshop
- Gregory D. Hager (1992). "Sensor planning for reactive robot programs," invited presentation at the Allerton Conference on Control and Computing
- Gregory D. Hager (February, 1993). "Efficient Solution of Large Systems of Nonlinear Constraints With Inexact Data and Explicit Termination Criteria," presentation at the Conference on Numerical Computation with Automatic Result Verification in LaFayette Lousiana
- Gregory D. Hager (May, 1993). "Task-Directed Computation of Qualitative Decisions from Sensor Data," presentation at IEEE Conf. on Robotics and Automation
- Gregory D. Hager (1993). "Solving Large Systems of Nonlinear Constraints with Application to Data Modeling," *Interval Computations*, in press
- Gregory D. Hager (1994). "Task-Directed Computation of Qualitative Decisions from Sensor Data," submitted to the *IEEE Trans. on Robotics and Automation*
- Gregory D. Hager, Sean Engelson, and Sami Atiya (1993). "On Comparing Statistical and Set-Based Methods in Sensor Data Fusion (with S. Engelson and S. Atiya)." *Proc. International Conference on Robotics and Automation.*
- Drew McDermott (1991) A reactive plan language. Yale Computer Science Report 864
- Drew McDermott (1991) Regression planning. *Int. J. of Intelligent Sys.* 6 (4), pp. 357–416.
- Drew McDermott (1991) "Robot Planning," Invited talk at National Conf. on Artifical Intelligence, July

- Drew McDermott (1991) Robot planning. Yale Computer Science Report 861
- Drew McDermott, (Nov. 1991). Invited presentation on "Transformational Planning of Reactive Behavior" at Ohio State University.
- Drew McDermott (1992) "Perceptual Confusion in Reactive Plans." *Proc. of the AAAI Spring Symposium.*
- Drew McDermott, (April 1992), Talk at University of Chicago on "Building and Fixing Diktiometric Maps for Robot Navigation."
- Drew McDermott, (April 1992). Talk at Northwestern University. "Transformational Planning of Reactive Behavior."
- Drew McDermott, (June 1992). Chaired panel on "Unified Theories of Planning." at the First International Conference on AI Planning Systems. U. of Maryland.
- Drew McDermott, (July 1992). Seminar on "Classical and Reactive Planning." Bolzano Summer School, Italy.
- Drew McDermott (1992). "Transformational planning of reactive behavior." Yale Computer Science Report 941
- Drew McDermott (1992) Robot planning. *AI Magazine*. Summer
- Drew McDermott (January 26–27. 1993). "Transformational Planning of Reactive Behavior," presentation at the MIT Workshop on Autonomous Underwater Vehicles, Cambridge, MA
- Drew McDermott (March 1993). "Planning is Automatic Programming for Agents," invited talk at the AAAI Spring Symposium
- Drew McDermott, (April. 1993). "Transformational Planning of Reactive Behavior." Brandeis University
- Drew McDermott (panel discussion at AAAI July. 1993) "Pros and Cons of Software Evaluation"
- Drew McDermott, William Cheetham, and Bruce Pomeroy (1991) Cockpit emergency response: the problem of plan projection. *Proc. IEEE Conf. on Systems, Man, and Cybernetics*. Charlottesville, Virginia

Final Report
ONR Contract Number N00014-91-J-1577
Principal Investigators: Drew McDermott and Gregory Hager
Yale University Department of Computer Science

3 Research Transitions And DoD Interactions

- Prof. McDermott did work with Bruce Pomeroy and Bill Cheetham of GE's Corporate Research and Development Laboratory on exporting some of the reactive planning ideas to the domain of planning for emergencies at crewstations. This work was reported at a paper that appeared in the proceedings of the IEEE SMC conference in October, 1991.

- Prof. McDermott was the General Chair of the First International Conference on AI Planning Systems, held in June of 1992. DARPA supplied some of the funds for this conference, which was viewed as the successor to the DARPA Planning Workshops.

- Prof. McDermott has been serving on the Technical Review Board for the ARPA/Rome Lab Transportation and Scheduling Initiative. The purpose of the board is to provide high-level feedback to researchers in this area, using insights gained from past research on planning and scheduling.

4 Software and Hardware Prototypes

1. We have exported the Reactive Plan Language interpreter (in Common Lisp) to the University of Washington and Georgia Tech.

2. The probabilistic time map has been split off from the planner, and is now available via anonymous ftp.

3. We have shared robotics software, including a constraint solver, visual tracker, and mobile-robot control software with interested institutions. This software is written in C and C++.