

If at First You Don't Succeed...

Kentaro Toyama and Gregory D. Hager

Department of Computer Science, Yale University, P.O. Box 208285
New Haven, CT, 06520

Abstract

One quality that makes biological systems appear intelligent is their robustness to difficult circumstances. Robustness is crucial to intelligent behavior and important to AI research. We distinguish between *ante-failure* and *post-failure* robustness for causal tasks. Ante-failure robust systems resist failure, whereas post-failure systems incorporate the ability to recover from failure once it happens. We point out the power of post-failure robustness in AI problems, closely examining one example in visual motion tracking. Finally, we raise theoretical issues and argue for greater effort towards building post-failure robust systems.

Introduction

How many of us have heard, “What happens if I do ... *this?*” usually accompanied by a mischievous grin and some flick of the wrist intended to wreak havoc on an otherwise “perfect” system? The ensuing moment involves us manually reinitializing a crashed system, while we mumble something about how the theory assumes that certain situations will not occur.

Robustness is a hallmark of intelligent behavior. It is an elusive quality that is ubiquitous in biological systems, yet rare in artificial ones. We can build systems like Deep Blue that challenge world champions in chess (Hsu *et al.* 1990). Yet, no system can reliably play the physical game of chess. Plenty of systems are able to move pieces from square to square in very constrained environments, but few systems, if any, can handle the differences in piece appearance from set to set. Even fewer systems would notice an accidentally fallen piece, pick it up, and return it to its proper square. None would recognize that a piece was missing and perform actions required (confront the culprit, ask for a replacement, find a substitute token, etc.) to continue the game. It is revealing that one of Deep Blue's human designers is required to act as a go-between in real games; the human mediator must perform all physical actions: report opponent moves, move pieces, and hit the clock.

The purpose of this paper is twofold. First, we define a new taxonomy with two categories of robustness — *ante-failure* and *post-failure* robustness. Briefly, ante-failure robustness focuses on avoidance and prevention of failure, whereas post-failure robustness emphasizes the need for recovery mechanisms once failure occurs. We review previous robustness research, and we outline the domain of tasks for which this distinction makes sense. The classification will make it easier to argue our second point, that while much has been accomplished in ante-failure robustness, little work, either practical or theoretical, has been done for its complement. We argue that post-failure robustness is a fundamental requirement of robust systems and present an example in visual tracking to back this claim. Our final aim is to encourage efforts to build post-failure robust agents.

Ante-Failure Robustness

Attempts to characterize robustness abound in statistics and engineering literature (Berger 1985). All of these definitions characterize types of *ante-failure* robustness: they quantify the ability of a system to resist failure in adverse conditions.

Ante-failure robustness analysis originates from efforts in statistics to develop robust estimators. Mention of robustness goes back to 1953 (Box 1953; Rey 1983): “... a statistical procedure is described as robust if it is not very sensitive to departure from the assumptions on which it depends” (Kendall & Buckland 1981). These departures from assumptions, some of which appear in the form of outliers, refer to observations which fall outside of the expected distribution of observations due either to unmodeled noise processes or pathological quirks of the observed system. Robust estimators are able to mitigate the effect of outliers and return reliable estimates of state in spite of them.

This statistical notion extends naturally to analogous concepts in decision theory and control theory. *Bayesian robustness*, which is informally described as “the sensitivity of [a] decision to assumptions ... that are uncertain” (Berger 1985), is concerned with possible errors or uncertainties in the specification of prior distributions or utilities and how they affect a decision.

*Copyright ©1997, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

In control theory, the question is whether a system built to achieve some goal can do so in the presence of different kinds of disturbances (Elbert 1984). A robust control system would be able to converge to its goal in spite of small disturbances. These concepts are intimately related. In some cases, good solutions in one domain transfer to another. For instance, if a control system attempting to keep a rocket upright exhibits instability due to outlier observations in its gyroscope circuitry, then a robust estimator that successfully ignores bad orientational data will add significant robustness to the entire control system.

We can thus characterize all ante-failure robustness work as asking the following question: “Given that the world differs from our expectations, can the system nevertheless operate within performance constraints?” Associated questions include “What kinds of deviations can the system withstand before it fails,” and “To what degree do input deviations result in output deviations?” As these questions indicate, ante-failure methods for robustness try to *prevent* failure.

Although ante-failure robustness remains an active field of research in the engineering community, many analytical tools have already been developed to quantify robust systems. Artificial intelligence has benefited from these advances, particularly in the areas of probability theory, utility theory, vision, robotics, and speech understanding. Post-failure robustness, however, remains largely unexplored.

Post-Failure Robustness

We loosely define post-failure robustness as the ability of a system to recover after failure.

Task Domains

“Robust” is an adjective often used to describe algorithms and systems. The ante-failure/post-failure distinction, however, makes sense only for a certain class of *tasks*. Although one might glibly remark that a system is post-failure robust, in fact, a system can only be said to be robust with respect to a particular task formulation. This subtle distinction will become clear as we describe the applicable domain.

Post-failure robustness is crucially related to causality as understood in signal processing. In signal processing, time-independent, *noncausal tasks* admit to a formulation in which all of the information about the problem can be given up front and for which the response by a task-solving system would be made after all input is given (Proakis & Manolakis 1996). At most, the task can have one deadline for responses, and it must be a hard deadline. Such a task is a “one-shot” problem where the solution is a single set of outputs or a single action. Failure in noncausal problems means unrecoverable failure. There is no “after” in which the system can correct itself. Therefore, “post-failure robustness” has no meaning for noncausal tasks.

In time-dependent, *causal tasks*, sequences of inputs are usually received in time, but more importantly, the nature of the task involves responses in sequence before all input is known. Response deadlines can be a result of the task requirements as in the case of real-time problems, or causality can appear as repeated interaction, as in the case, for example, of a command-line interpreter. Post-failure robustness makes sense only for causal tasks.

Since “recovery” after a failure suggests that failure is not catastrophic, we only consider *tasks of policy* or tasks which include tasks of policy as necessary sub-tasks. Tasks of policy are those tasks where some sort of failure is locally defined for a time instant, but for which isolated failures do not necessarily imply catastrophic failure of an overall task.

As mentioned above, post-failure robustness applies only to a specific task definition. A good example of a task of policy is a visual sensing task, disembodied from higher-level tasks: “Observe the position of an object to within 1cm for as much of the time as possible.” Implicit in this task definition is a definition of failure: a system fails if it cannot determine position accurately. The temporary failure to perform this task for some time interval, however, does not necessarily imply an utter failure to follow policy. Also, even though an algorithm may be post-failure robust in performing this particular task, the same algorithm may be non-robust with respect to another task. One can imagine tasks where even minor temporary failures may prevent the achievement of an end goal.

Therefore, the problems that concern post-failure robustness are those tasks of policy where failure to observe policy temporarily does not result in catastrophic failure.

Motivation

Computer science as a broad field offers some examples of post-failure robust systems. From the outset, we ask students to make their projects “robust.” By this, we mean that their programs should respond in a reasonable manner regardless of the input. In the systems community, robustness, or reliability, is the primary focus (Siewiorek & Swarz 1982). As early as 1952, von Neumann was concerned with “the synthesis of reliable organisms from unreliable components,” by which he meant the construction of robust computational black boxes out of imperfect hardware parts (von Neumann 1956). Since then, systems researchers have sought both ante-failure and post-failure mechanisms to hide hardware problems from the end user.

In AI applications, however, difficulties come not only from inadequate debugging or from hardware errors, but also from unexpected input and from the wide array of executable behaviors. Robustness for AI is simply choosing the best action despite complex or overwhelming input. The difficulties lie in extracting the relevant information from the environment and

choosing actions based on knowledge that is often incomplete.

There are several practical reasons to pursue post-failure robustness over ante-failure robustness. We give three: (1) Ante-failure robustness must deal with different modes of failure, but a single post-failure procedure often suffices to recover after any type of failure. (2) Ante-failure robustness methods often reduce performance in ideal conditions. (3) Perfect ante-failure robustness is impossible to achieve in a complex world; if failure is inevitable, one should plan for recovery.

For any given domain, failure has many causes. In speech understanding, for instance, speaker-induced noise, mumbling, or loud environmental noise may all cause an inability to comprehend what was said. The ante-failure approach to dealing with these problems might include separate filters and compensatory mechanisms to handle each source of difficulty. Even if the components worked perfectly without interfering with one another, it is still possible that some other problem, such as a slip of the tongue, is not accounted for. The post-failure approach to all of these problems is a single speech act that we often employ ourselves: "Excuse me, what did you say?" So, where many ante-failure devices are required to avoid failure altogether, a single post-failure mechanism delivers, in one stroke, enough power to recover regardless of the reason for failure.

Another difficulty with ante-failure methods is their tendency to degrade the performance of a system under ideal conditions. For example, one consequence of having to handle many failure modes with an ante-failure approach, as described in the previous paragraph, is that it incurs additional processing costs. Monitoring for different kinds of deviations will require separate checking for each dimension of deviation. Another, perhaps more interesting, example is that statistical robust estimators lose precision on good data. The better modeled the data, the worse robust estimators perform when compared with optimal estimators (Berger 1985).

Finally, in most domains, perfect failure prevention is impossible. Causal domains are usually associated with uncertainty about the environment, the very thing that makes many AI domains interesting. For example, in robot navigation, there are many types of obstacles a mobile robot may encounter in going from Point A to Point B. Walls, doors, stairs, trash cans, elevators, people, pets, and other robots, are just a few examples of these. Existing robots can avoid some subset of these obstacles and research continues on the rest, but even assuming all of these particular problems were solved, could a mobile robot then deal with a thin, low-hanging clothesline? Does a specific module need to be created for this new type of obstacle? And if one were created, would it then make the system perfectly collision-free? Probably not. "To err is human," it is said, but erring extends beyond our species. *Making*

mistakes is an unavoidable consequence of executing complex behaviors in a complex world.

Intelligent action does not end with failure to perform a task perfectly. In the face of failures, an intelligent agent ought nevertheless take the best possible action to fulfill some higher goal. Agents should be expected to do the right thing to correct themselves even when things go wrong. This is the essence of post-failure robustness.

An Example

As an example, we examine research in real-time visual tracking. The main thrust of this community has traditionally been in ante-failure robustness, largely because of its historical ties to engineering. Post-failure techniques can be surprisingly effective, however, and below, we describe such work.

Problem and Previous Work

Visual tracking is a task of policy, where a system tries to compute, for each time instant, the state of a target object as it moves in real-time, using live video data. The state includes quantities such as position, shape, velocity, and so forth. Failure for a particular time instant is the inability to deliver complete and accurate state information. Although there are many algorithms for visual tracking (using intensity-based templates, color histograms, edge-based contours, etc.), all experience problems with visual perturbations. Changes in ambient illumination, distractions in the background, foreground occlusions, and erratic motion are a few of the typical problems.

Researchers have expended significant time and effort trying to handle these problems from an ante-failure perspective. Distractions, or objects similar to the target, can be ignored by processing only a narrow window of pixels surrounding the target (Hager & Toyama 1996a; Vincze 1996), or using foveation to effectively blur the image region around the target (Burt & van der Wal 1990; Terzopoulos & Rabie 1995). Occlusions, where opaque objects intercept the camera's line of sight to the target, can be handled by robust matching techniques (Gennery 1992; Hager & Belhumeur 1996; Lowe 1992; Kosaka & Nakazawa 1995) or with dynamic state prediction (Blake, Curwen, & Zisserman 1993; Rizzi & Koditschek 1993; Terzopoulos & Szeliski 1992). Changes in ambient lighting are often handled by concentrating on color cues (Kahn *et al.* 1996; Wren *et al.* 1995). And, fast or unpredictable motion can be attacked with combinations of faster hardware, full-frame processing (Burt 1988; Nishihara 1996), or probabilistic dynamics (Isard & Blake 1996).

Some have tried to make visual tracking ante-failure robust with respect to all of these problems (Isard & Blake 1996; Toyama & Hager 1995), but none have succeeded for two reasons. First, the component solutions to the problem are not always reliable. For ex-

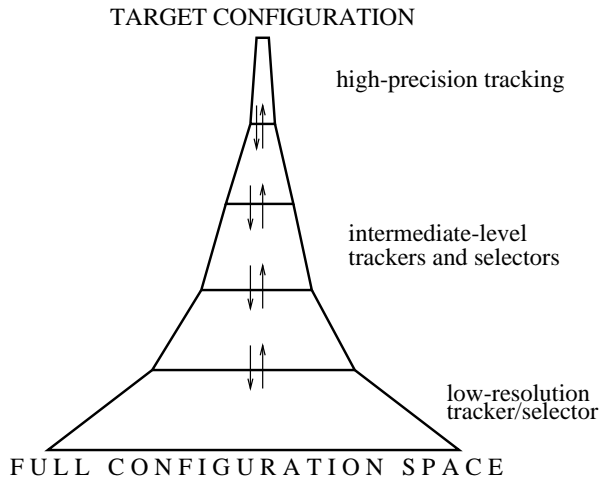


Figure 1: The Incremental Focus of Attention (IFA) framework. Selectors search for the target object using focus of attention heuristics. Trackers actually track the target once locked on to it.

ample, a visual tracking algorithm cannot be expected to accurately track a face wholly occluded by a wall. Second, the problems are not orthogonal, making solution composition difficult. Although narrow focus may help to avoid distraction, for instance, it is entirely unsuited for dealing with unpredictable object motion (imagine trying to follow a butterfly through a telescope). Ante-failure methods are therefore inadequate. A post-failure mechanism for recovering track of a lost target is necessary.

Incremental Focus of Attention

The *Incremental Focus of Attention* (IFA) framework begins to fill this gap using post-failure methods. Because most of this work is reported elsewhere (Toyama 1996; Toyama & Hager 1996), we offer only a brief description.

The framework combines multiple tracking schemes, using them as attention focusing mechanisms which rapidly find a target object in an image. Each tracking algorithm is assigned to a *layer* in the framework’s hierarchy (see Figure 1). Algorithms are modified, if necessary, to diagnose their own success with tracking. Layers are sorted by their output precision, with more precise algorithms at the top and less precise algorithms at the bottom. Higher layers are usually slower and spatially focused, while lower layers are faster and have broader perspective.

Under ideal tracking conditions, with constrained object velocity, steady ambient lighting, lack of occlusion, and so on, tracking proceeds at the topmost layer, and the system as a whole delivers complete state information. As tracking conditions deteriorate, control cedes to lower layers which are able to continue track-

ing only some of the state parameters. When tracking conditions improve, control returns to the higher layers by using partial information gained at lower layers to constrain the search procedure in higher layers. Under severe visual disturbances, such as full occlusion, tracking fails completely, but through repeated attempts to find the object, tracking recovers when ideal tracking conditions return.

The IFA approach to tracking a person’s face, for example, is as follows: The lowest level searches the entire image for motion, when it exists, or for flesh-colored pixels. Middle layers track approximate position and orientation of flesh-toned “blobs.” The top layers use a correlation-based algorithm to track with sub-pixel accuracy. If the target face moves slowly enough, without occlusion, and under steady lighting, the top layer tracks the intended face at full precision. When the face moves too quickly for the top layer to track, middle layers take over, offering approximate state information. At any point, if tracking fails, control falls to the lowest layers, where a multi-layered search is performed until the face is found, at which time control ascends again to the topmost layer.

The IFA face tracking system has been demonstrated many times both in and out of its normal environment and has survived rigorous testing by skeptical audiences (Hager & Toyama 1996b). The subjective experience of watching the system in action is that it is extremely robust. Although no claims are made about any similarities to biological vision systems, IFA was inspired by mammalian visual strategies, and some observers have noted that the behavior of the tracking system seems almost human, both in method and robustness: While unable to resist temporary failure during occlusions, for example, the IFA system notices mistracking, makes serial saccadic searches for the target, and gracefully recovers after the disturbances cease. The considerable robustness that comes in addition to the ante-failure robustness of the component tracking algorithms is a result of the relatively simple post-failure mechanisms in the framework itself. For more detail about IFA, see (Toyama 1996; Toyama & Hager 1996).

A Comparison

We now compare the behavior of an ante-failure, occlusion resistant tracker with our face tracker’s response to occlusion. As our ante-failure example (**AF**), we use an SSD-based correlation technique which detects and factors out occlusions by ignoring outliers — pixels which are significant mismatches in an otherwise well-correlated image (Hager & Belhumeur 1996). **AF** is an example of a standard ante-failure approach: methods from robust statistics are applied to filter out non-signal data. For our post-failure example (**PF**), we use the face tracker described above. We consider the case when the top-layer is the same SSD tracker without occlusion resistance.

When there is no occlusion and when tracking conditions are ideal otherwise, both trackers operate with subpixel precision, but **AF** is 15–20% slower due to the extra computation required to detect the outliers. This computation must take place even in the absence of occlusion. **AF** is also somewhat less effective at tracking the non-occluded face, requiring more cycles to converge to the target. The latter effect arises from competing pressures to explain image differences as either occlusions or motion. These points illustrate some of the drawbacks of ante-failure robustness techniques discussed earlier.

Experiments with occlusion also show how the two tracking algorithms differ. During a small partial occlusion, *e.g.*, a hand scratching the face, **AF** continues to track with negligible loss of precision, effectively screening out the occluder. When the occlusion ceases, tracking continues as before. **PF**, on the other hand, reverts to color-based blob tracking, with significant loss of precision in determining position and orientation. When occlusion ends, full-precision tracking is recovered within 100 msec. This is the one instance in which **PF**'s performance compares poorly with **AF**.

In instances when occlusions obscure over 40–50% of the face, **PF** outperforms **AF**. **AF** is completely unable to track the face, and mistracking is permanent, even after occlusions cease. This is an example of the inability of ante-failure robust techniques to handle extremes of input. **PF**, on the other hand, displays the same behavior as it did for small partial occlusions. If a complete occlusion occurs (when no skin is visible), **PF** begins a multi-modal search that recovers when the face reappears. Recovery takes between 150 msec to several seconds, depending upon the number of distractors in the background. Thus, without explicit tailoring to handle occlusion, **PF** is nevertheless able to recover from failure due to occlusion.

Needless to say, **AF** is completely unable to continue tracking if the face undergoes other visual disturbances, such as rapid motion or sudden lighting changes. **PF**, while not able to track through these disturbances, recovers tracking when they end.

Lastly, we reiterate that although we have chosen to separate ante-failure and post-failure mechanisms in this example for the purposes of comparison, it is easy to combine the two to create a system that exhibits both ante-failure and post-failure robustness.

Other Robust Systems

The IFA paradigm extends beyond vision into robotics. For example, the current IFA system can be used in visual servoing applications.

An example of a visual-servoing task is to servo a screwdriver onto a screw. Although this task involves a single goal, it also includes subtasks of policy: to maintain visual track of the screwdriver and to maintain grasp of the screwdriver. The former problem can be handled by an IFA tracking system which outputs

the position of the screwdriver along with an additional variable indicating tracking reliability to the servoing procedure. If the tracked state is not recoverable at the precision desired by the servoing algorithm, the servoing algorithm instructs the manipulator to stop. When tracking recovers, so does servoing. The problem of holding the screwdriver can be solved by having the tracking system notice sudden downward motion by the screwdriver indicating that the manipulator has lost grasp of the screwdriver. With proper post-failure recovery algorithms to pick up a dropped tool, the eventual goal can be attained.

Robotics is filled with similar tasks which admit to solutions that are forgiving of temporary policy failures. What about collision recovery in addition to collision avoidance? Or path exploration and replanning instead of provably-correct path planning? These should all be pursued with post-failure robustness in mind.

Some previous work in AI could be classified as post-failure robust, as well. Lessons common to this work may provide a springboard for further development.

The planning literature has perhaps been the most actively involved in post-failure robustness. Two different paradigms were recognized as important in early planning work. *Conditional planning* attempts to account for possible action failures by planning for just such contingencies (Fikes, Hart, & Nilsson 1972), and *replanning* allows for adjustment or reconstruction of plans when execution deviates from an original plan (Ambros-Ingerson & Steel 1988). We hope to see this research continue, and more importantly, teach relevant lessons to the rest of the AI community.

More recently, behavior-based systems display post-failure robustness. Behavior-based robot systems appear to possess an “emergent” intelligence, running high-level behaviors such as empty can collection when possible, and running low-level behaviors such as obstacle avoidance and aimless wandering at other times (Brooks 1986). Even if the desired high-level activity is not carried out successfully, these robots are able to survive in their environment. Post-failure robustness is achieved by waiting out difficult circumstances with simple behavior.

Finally, the literature in reinforcement learning has always taken for granted that agents will be able to recover from failures (Kaelbling 1990; Watkins 1989). Still, most of this work sidesteps the issue of recovery itself. In some instances, the recovery is performed by the researchers themselves (Michie & Chambers 1968), and in others, the work is performed entirely in an virtual environment, where “recovery” only requires reinitialization of abstract data (Furuta, Ochiai, & Ono 1984). Learning from trial and error, though ultimately dependent on post-failure recovery, is therefore a separate problem. Non-learning systems can be extremely robust, as many lower animals are, and systems capable of learning can be very brittle, as many

existing robots are. Thus, it is important to isolate the problem of recovering from failure and to distinguish it from related areas of research.

The Challenge

We want to encourage more work in post-failure robustness, *per se*. The traditional credo of AI is to build systems that work first, and to analyze how and why second. We suggest a vigorous return to this method, especially with respect to designing robust systems. There aren't enough robust systems (whether ante- or post-failure) for anyone to draw meaningful conclusions about achieving robustness in general.

Certain characteristics appear common to robust systems. We would like to see these qualities in more systems:

- Self-monitoring and cognizant failure.
- Partial or full operation under non-ideal circumstances (ante-failure, safety net techniques).
- Reinitialization (starting over) or recovery (continuing) after failure.
- Application of knowledge gained from failure to future operation (adaptive recovery, reinforcement learning, etc.).

Our current lack of formal tools for analyzing robustness should not hinder us from developing systems that work. On the other hand, without analytical tools to evaluate post-failure robustness, research is inhibited. Results are difficult, if not impossible to analyze, and little comparison can be made to other work. Therefore, to theorists, we issue a challenge: How can we formalize post-failure robustness? Any formalism should address the following questions:

- Can we establish a theory of post-failure robustness abstracted from specific problems?
- How can we formalize the environmental assumptions relevant to system robustness?
- What is the value of cognizant failure over positive (non-cognizant) failure?
- How can we compare two systems with respect to post-failure robustness?
- Is there a meaningful way in which ante-failure and post-failure methods can be compared?
- Can we “factor out” differences in hardware between systems, or is it necessary to define *bounded robustness*, in a spirit similar to bounded optimality (Russell & Wefald 1991)?

Developing a theory of robustness is difficult because of its context-dependent nature. The same algorithm may appear robust or brittle depending on who observes it, what environment it operates in, when it is executed, and how it is used by a higher-level application. Nevertheless, we believe there is a basis for our intuitive notion of robustness that can be formalized.

Objections

Finally, we anticipate and address possible objections to pursuing post-failure robustness in practice and theory.

One objection is a tacit resistance among academic researchers to working on robustness for its own sake. We often feel that making a system robust is easy or that the tedium does not result in interesting new work. Recovering from failure, however, is rarely easy or uninteresting, especially in causal domains. In addition, elegant solutions in one domain may apply to other types of problems.

Other objections stress the importance of ante-failure work. A fully automated pilot, for example, might hold human life in its hands. Recovery after the failure of a crash is not an option. In such problems, ante-failure robustness must be pursued thoroughly. Likewise, recovery from failure is useless if there are no algorithms to recover *to*. Asking a Spanish-speaker to repeat a request would not help at all if our natural language system does not understand Spanish. To these objections, we reemphasize the necessity of work in robustness overall. Post-failure robustness does not replace ante-failure robustness, it complements it. Neither should be pursued to the exclusion of the other.

Conclusion

In the late 18th century, a Hungarian baron, Wolfgang von Kempelen — who might, in ambition at least, be considered a precursor of the Deep Blue team — built a chess-playing machine that toured the world, stunning audiences with its ability to play a reasonable game of chess. Kempelen's machine interested Edgar Allan Poe, and in 1836, Poe wrote an essay in which he surmised that the “machine” actually hid a person inside. Poe guessed correctly, but based much of his reasoning on the fallacy that a machine would never lose (Poe 1836). As computer scientists, and especially as artificial intelligence researchers, we feel immune to this fallacy. After all, we have seen robots crash into walls and translation programs spout gibberish. But, in our hearts, we blame ourselves for the faults of our systems. We feel that in due time, we will be able to build perfect collision-avoiding robots and error-free translators. Apparent errors, however, are inevitable in a complex environment. We must learn to accept “mistakes” as inherent in our discipline and move on to endowing artificial agents with post-failure, as well as ante-failure, robustness.

Acknowledgments

This research was supported by Army DURIP grant DAAH04-95-1-0058, by National Science Foundation grant IRI-9420982, and by funds provided by Yale University. Discussions with Zachary Dodds provided valuable insight. We are grateful for the comments by the three anonymous reviewers, all of which were helpful during paper revision. We are especially appreciative of one reviewer's exceptionally encouraging remarks.

References

- Ambros-Ingerson, J., and Steel, S. 1988. Integrating planning, execution and monitoring. In *Proc. 7th Nat'l Conf. on AI*, 735–740.
- Berger, J. O. 1985. *Statistical Decision Theory and Bayesian Analysis*. New York, NY: Springer-Verlag.
- Blake, A.; Curwen, R.; and Zisserman, A. 1993. Affine-invariant contour tracking with automatic control of spatiotemporal scale. In *Proc. Int'l Conf. on Comp. Vision*, 421–430.
- Box, G. E. P. 1953. Non-normality and test on variances. *Biometrika* 40:318–335.
- Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE J. of Rob. and Autom.* 1(1):24–30.
- Burt, P., and van der Wal, G. 1990. An architecture for multiresolution, focal, image analysis. In *ICPR90*, 305–311.
- Burt, P. 1988. Attention mechanisms for vision in a dynamic world. In *ICPR88*, 977–987.
- Elbert, T. F. 1984. *Estimation and Control of Systems*. New York: Van Nostrand Reinhold.
- Fikes, R. E.; Hart, P. E.; and Nilsson, N. J. 1972. Learning and executing generalized robot plans. *Artificial Intelligence* 3(4):251–288.
- Furuta, K.; Ochiai, T.; and Ono, N. 1984. Attitude control of a triple inverted pendulum. *International Journal of Control* 39(6):1351–1365.
- Gennery, D. B. 1992. Visual tracking of known three-dimensional objects. *Int'l Journal of Computer Vision* 7(3):243–270.
- Hager, G., and Belhumeur, P. 1996. Occlusion insensitive tracking of image regions with changes in geometry and illumination. Technical Report DCS-TR-1122, Yale University.
- Hager, G., and Toyama, K. 1996a. XVision: A portable substrate for real-time vision applications. In *Proc. ECCV*, volume 2, 507–512.
- Hager, G., and Toyama, K. 1996b. XVision: Interaction through real-time visual tracking. In *CVPR Demo Program*.
- Hsu, F.-H.; Anantharaman, T.; Campbell, M.; and Nowatzyk, A. 1990. A grandmaster chess machine. *Scientific American* 263(4):44–50.
- Isard, M., and Blake, A. 1996. Contour tracking by stochastic propagation of conditional density. In *ECCV96*, I:343–356.
- Kaelbling, L. P. 1990. Learning functions in k-DNF from reinforcement. In *Machine Learning: Proc. 7th Int'l Conf.*, 162–169.
- Kahn, R.; Swain, M.; Prokopowicz, P.; and Firby, R. 1996. Gesture recognition using Perseus architecture. In *CVPR96*, 734–741.
- Kendall, M. G., and Buckland, W. R. 1981. *A Dictionary of Statistical Terms, 4th Edition*. Edinburgh: Oliver and Boyd.
- Kosaka, A., and Nakazawa, G. 1995. Vision-based motion tracking of rigid objects using prediction of uncertainties. In *Proc. 1995 IEEE Conf. on Rob. and Autom.*, 2637–2644.
- Lowe, D. G. 1992. Robust model-based motion tracking through the integration of search and estimation. *Int'l Journal of Computer Vision* 8(2):113–122.
- Michie, D., and Chambers, R. A. 1968. BOXES: An experiment in adaptive control. In Dale, E., and Michie, D., eds., *Machine Intelligence 2*. Amsterdam: Elsevier. 125–133.
- Nishihara, H. K. 1996. Real-time vision. In *CVPR Demo Program*.
- Poe, E. A. 1836. Maelzel's chess player. *Southern Literary Messenger* 2(5):318–326.
- Proakis, J. G., and Manolakis, D. G. 1996. *Digital Signal Processing*. Prentice Hall.
- Rey, W. J. J. 1983. *Introduction to Robust and Quasi-Robust Statistical Methods*. New York, NY: Springer-Verlag.
- Rizzi, A., and Koditschek, D. E. 1993. Further progress in robot juggling: The spatial two-juggle. In *Proc. 1993 IEEE Int. Conf. on Rob. and Autom.*, 919–924.
- Russell, S., and Wefald, E. 1991. *Do the Right Thing: Studies in Limited Rationality*. Cambridge, MA: MIT Press.
- Siewiorek, D. P., and Swarz, R. S. 1982. *The Theory and Practice of Reliable System Design*. Bedford, MA: Digital Press.
- Terzopoulos, D., and Rabie, T. F. 1995. Animat vision: Active vision in artificial animals. In *Int'l Conf. on Comp. Vision*, 801–808.
- Terzopoulos, D., and Szeliski, R. 1992. Tracking with Kalman snakes. In Blake, A., and Yuille, A., eds., *Active Vision*. Cambridge, MA: MIT Press.
- Toyama, K., and Hager, G. D. 1995. Tracker fusion for robustness in visual feature tracking. In *SPIE Int'l Sym. Intel. Sys. and Adv. Manufacturing*, volume 2589.
- Toyama, K., and Hager, G. 1996. Incremental focus of attention for robust visual tracking. In *Proc. CVPR*, 189–195.
- Toyama, K. 1996. Handling tradeoffs between precision and robustness with incremental focus of attention for visual tracking. In *Working Notes AAAI Symp. on Flexible Computation in Intelligent Systems*, 142–147.
- Vincze, M. 1996. Optimal window size for visual tracking. In *ICPR96*, A91.1.
- von Neumann, J. 1956. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Annals of Mathematics Studies* 34:43–98.
- Watkins, C. J. 1989. *Models of Delayed Reinforcement Learning*. Ph.D. Dissertation, Cambridge University, Cambridge, U.K.
- Wren, C.; Azarbayejani, A.; Darrell, T.; and Pentland, A. 1995. Pfinder: Real-time tracking of the human body. In *Vismod*.