Lightweight Vision
-or-
How I Learned to Stop Worrying and Love My Camera

Ian Horswill
MIT AI Laboratory
ian@ai.mit.edu

Introduction

Sensing is often the limiting factor in mobile robot performance, particularly for robots that operate in dynamic environments. Both the robot's opportunities for behavior and its ability to respond to contingencies are limited by the breadth and reliability of its percepts.

While it can be difficult to get people to admit to it in writing, there is a common attitude in both the AI and robotics communities that vision is not even worth considering as a sensor because of its expense and unreliability. I will refer to this as "fear of vision." Vision has had a particularly bad reputation in some communities because of the huge volume of data which it produces (a single full-resolution frame grabber produces about 27MBytes/second) and the complex, iterative, floating-point computations required to interpret the data. Vision is, of course, often expensive by necessity. If one is faced with the task of constructing digital terrain maps from high resolution satellite images, then the only option is to build a full stereo or shape from shading system and use whatever computational resources are necessary to do the job. If one only has to make a robot drive down the corridor however, then one has many more options. A vision researcher might wish to use an expensive technique because of its intrinsic research interest. An engineer building a simple vacuum cleaning robot, however, will presumably be content with anything that does the job. Many robot builders have put up with sensors which give them less information than they need, simply because they felt that vision was impractical.

I believe that it is practical to build very simple and reliable vision systems to perform a variety of tasks. Over the past five years I have developed a number of simple vision systems for piloting mobile robots in real time by taking advantage of the structure of robot's task and environment. The most recent of the systems, the Polly system (see Horswill [5][6]), gives simple tours in an unmodified office environment using real-time vision processing performed by an inexpensive on-board computer. The robot was build for less than $20K. Gavin and Yamamoto have recently developed a delivery platform which we hope will be able to run much of the algorithms from Polly for less than $1K (see Gavin and Yamamoto [4]). Such a system is sufficiently inexpensive that it can realistically be incorporated into mass-market consumer products. To date, we have ported the low level navigation code from Polly to the platform and used it to control an IS Robotics R2 robot.

Polly's vision algorithms operate on low resolution (64×64 or less) images and involve no iterative optimization computations whatsoever. In fact, there are not even any floating point computations.

Lightweight vision

There are a number of resources that the designer can use to simplify the structure of a vision system. One such resource is the structure of the agent's task. It tells the designer what information is needed and what performance is required. More importantly, it tells the designer what information is *not* needed and what performance is *not* required. Computing more information means computing more representations, which is obviously more expensive, or squeezing more information into the existing representations, which is often even worse. The first principle taught in most undergraduate AI classes is that a good representation makes the important information explicit, and nothing else. As one squeezes more and more information into a representation, the information tends to become less and less explicit, meaning that further, non-trivial processing is required simply to look at the representation and extract the information which was needed in the first place. Improving performance parameters generally means making trade-offs, usually either trading cost for performance, or trading one performance characteristic for another. Improving unimportant performance parameters is therefore not only wasted effort, it is sometimes a waste both of money and of important performance parameters

Resolution is a useful case in point. Many researchers I have talked to have taken it for granted that images below 128×128 are mostly useless, whereas Polly uses resolutions as low at 16×12. Polly demonstrates that surprisingly good results can be obtained with surprisingly low resolutions (see Horswill and Brooks [7] and

Pomerleau [10] for other examples). Obviously, some tasks and environments require higher resolution, but many do not. Even when it is necessary to sample one modality at high resolution, it may be possible to sample the other modalities at much lower resolution. If a smooth, finely textured object moves past the camera, then the intensity field will vary rapidly in both space and time and so will have to be finely sampled to estimate the motion field. The motion field itself however, will vary much more slowly and so may not need to be sampled nearly so finely.

The structure of the environment is another useful resource. The presence of simplifying structures in the environment often allows simple computations to be substituted for expensive ones. The fact that the floor in Polly's environment is textureless means that a texture detector can be used as an approximate floor detector. This allows Polly to rapidly separate obstacles from the background using an edge detector. This simplification can be done in a principled manner by making the special structure of the environment (the floor's appearance) explicit and describing the simplification in the form of a general lemma (see [6]). One advantage of the lemma is that it allows other people to perform the same optimization in the future. Another is that the lemma shows what is important about the simplification and what is not. In the case of obstacle detection, it makes explicit the conditions under which the substitution can be made and the class of texture detectors which can be used.

A final resource available to the vision system designer is the use of multiple strategies in parallel. Using multiple specialized systems with independent failure modes for a single task can increase generality and improve reliability. The follower system in [7] uses two different strategies for tracking a moving object. When one fails, the other takes over. Multiple strategies can also be used for similar tasks with differing performance constraints to avoid having to build a single system which satisfies both sets of performance constraints. Polly uses separate edge detectors for detecting the vanishing point of a corridor and for figure-ground separation. This allows the edge detectors to be tuned separately in scale and orientation space, and so simplifies overall system design.

Conclusion

This proposal falls within the basic task-based/active vision approach which recently has received a great deal of attention (see Aloimonos [1], Bajczy [2], Ballard [3], Horswill [8], and Ikeuchi [9]). In this paper, I have focused on cost considerations. My major point is that a researcher need not invest $100K in Datacube boards to use real-time vision. Much more modest computers often suffice if the right method can be found.

It is vital that vision be made cheap if it is ever to leave the laboratory. In the past, the cost of the necessary hardware has prevented vision not only from being used in consumer products, but even from being used by researchers not specifically doing vision research.

Practical consumer robots will have to use large sensor suites and run on relatively little power. While it is unlikely that a practical consumer robot could be built using only vision, it is equally unlikely that one could be built using only tactile and sonar data. A low cost vision system could greatly improve the performance of a consumer robot such as a vacuuming robot. Vision can allow the robot to efficiently and accurately orient itself with respect to a wall, to follow a wall, to avoid non-geometric hazards, and to determine the approximate color and texture of a rug.

I believe that by using the structure of a robot's task and environment, we can often produce very simple vision systems, systems economical enough to be mass produced as consumer products or to be incorporated into software for the next generation of multi-media computing machines.

References

[1] John Aloimonos. Purposive and qualitative active vision. In *DARPA Image Understanding Workshop*, 1990.

[2] Ruzena Bajcsy. Active perception vs. passive perception. In *Proc. Third IEEE Workshop on Computer Vision: Representation and Control*, pages 55–59. IEEE, October 1985.

[3] Dana H. Ballard. Animate vision. *Artificial Intelligence*, 48(1):57–86, 1991.

[4] Andrew S. Gavin and Masaki Yamamoto. A fast, cheap, and easy system for outside vision on mars. In *to appear in Intelligent Robots and Computer Vision XI*, Cambridge, MA, September 1993. SPIE.

[5] Ian Horswill. Polly: A vision-based artificial agent. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 824–829. AAAI, MIT Press, 1993.

[6] Ian Horswill. *Specialization of perceptual processes.* PhD thesis, Massachusetts Institute of Technology, Cambridge, May 1993.

[7] Ian Horswill and Rodney Brooks. Situated vision in a dynamic environment: Chasing objects. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, August 1988.

[8] Ian D. Horswill. Reactive navigation for mobile robots. Master's thesis, Massachusetts Institute of Technology, June 1988.

[9] Katsushi Ikeuchi and Martial Herbert. Task oriented vision. In *DARPA Image Understanding Workshop*, 1990.

[10] Dean A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1), 1991.