Preface

The work described in this book is the result of a sequence of experiments that began with the idea of building an honest real-time (60 Hz) vision system. The sight of a slick, high-tech robot plodding along at a snail's pace has always rubbed me the wrong way. John Jarvis originally pointed out that moments were a simple enough, and well enough understood, computer vision operation that we really ought to be able to do them faster. They seemed well suited to making a vision system that was fast enough to serve as a sensor for a fast robot, even if the scenes had to be quite simple. I set off to make a vision system that would process simple scenes quickly, rather than process more complex scenes slowly, as did the majority of the vision and robotics community. I could then investigate tightly coupled robot/vision systems.

Two years later, I had a working vision system, and in the meantime had created a system to control a PUMA 260 robot in C. Suddenly, the need for a demonstration of the potent capabilities of the work was upon me. For reasons no longer clear, but doubtless related to the simplicity of the resulting image, I threw together a system to catch a ping-pong ball rolling in 2-D across a table. The ping-pong ball had the advantage of being very white, and the ball would break rather than allow the robot to come to harm.

After a subsequent explosion in the complexity of the code, the robot caught most of a certain class of trajectories with high reliability. It would even catch balls bounced off a piece of foam shortly before they were to be caught, amply demonstrating the true real-time nature of the system. It was clear, however, that catching the remainder of the balls would require more subtle strategies not easily captured in my program — it was likely that even trying would cause the whole thing to collapse under its own weight. Nonetheless, the demonstration was a vast success, and the volume of demonstrations prevented me from doing much additional work on the system, affording me the time for thought.

Since I could catch in two dimensions, it only stood to reason that I should be able to do it in three. About this time, I heard of the challenge/contest announced by Professor John Billingsley of Portsmouth Polytechnic (U.K.) for robot ping-pong, and since I had already thought along these lines, Professor Billingsley's proposal provided an excuse for further ruminations. (Many thanks to Professor Billingsley for his well-thought-out rules.)

Some time later, I heard a rumor that Dan Kodeschek of Yale had thought of catching objects. I put off all consideration of ping-pong, and rushed to create a 3-D catcher, since I knew it should be straightforward. The resulting system proved able to catch balls in a styrofoam coffee cup, exhibiting some spectacular behavior due to primitive error recovery code.

By this time, however, it was clear that I was going about this entirely the wrong way. Alarmingly, it appeared that if I attempted to extrapolate the system to full ping-pong, it was never going to work. There was not going to be a magic algorithm that could compute an ideal ping-pong player's response, as had been the case (at least in practice) in the previous demonstrations. Trying to address the different cases was going to result in a nonfunctional morass; the line between when to apply one strategy and when to apply another was thin, bordering on nonexistent.

What I had to do was to capture the subjective skills of a trained human. A person's skilled techniques appeared then, and now, equally applicable to a person playing ping-pong, or to a person working on an assembly line. In both cases, people make many clever motions, without seeming to expend much thought, and one can improve their performance by giving small hints such as "keep your elbow up." By contrast, a robot would need a whole new algorithm.

This book describes an experiment to try to capture this type of skilled behavior, resulting in a robot ping-pong player which can play and beat human players. Along the way to a working system, I discovered and had to solve many interesting problems. The vision system, the expert robot controller, the low-level robot controller, and the overall system design were significantly affected by the need to make the system respond accurately in a dynamic environment. By working only with static problems, the vision and robotics communities have not yet had to understand dynamic environments.

Accordingly, this book begins to pave the way. At least one chapter is devoted to each of the four major subsystems above; each chapter describes what the subsystem must accomplish, how it does so and why, and quantitatively documents its performance. Although the book has the technical content of a work oriented at robotics professionals; computer scientists, physicists, hobbyists, and the technically curious should be able to read the book and gain insight into how the system works and is put together — and an appreciation of how good the human machine really is.

The management of AT&T Bell Laboratories, and especially John Jarvis, deserve immense credit for seeing the technical merit behind a project of little instantaneous gain. I think this characteristic reflects extremely well on the quality of the institution.

I am indebted to many people for their aid in the successful completion of this work. Many thanks to Professor Richard Paul (University of Pennsylvania), for his support, encouragement, immense practical experience, and especially, wise judgement. To Professor Ruzena Bajcsy (Penn) I owe the emphasis of carrying through ideas to working results, thus eliminating the need and possibility of trying to justify ideas by hand-waving. Professor Rodney Brooks and Professor Tomás Lozano-Pérez (both of MIT) have my thanks for their suggestions on how to improve the book's readability.

Thanks to George Whyte and Richard Seide, the able mechanical and electronic construction crew. Thanks to Ray "Plot" Soneira and his indomitable ray_plot, which supported the computer graphics in this book. I am indebted to the creators of the MEGLOS multiprocessor system and the MULGA VLSI design system: Bob Gaglianello, Howard Katseff, Krish Ramakrishnan, Beth Robinson, Brian Ackland, Jay O'Neill, and Neil Weste. Thanks are also due my other colleagues at the Labs who were always ready with ideas of strategies and effects the system ought to deal with, and put up with the sound of ball against table, paddle, and wall, and the screams from man and machine as the latter attempted autorobocide at spectacular speeds. Thanks for hanging in there, King Pong.

Finally, many, many thanks to my wife, Amy, who provided marathon-level encouragement, and mental, moral, physical, and logistical support.

Holmdel, New Jersey
November, 1987

R.L.A.