Machine Intelligence and Knowledge Engineering for Robotic Applications

Edited by

Andrew K. C. Wong

Alan Pugh

Machine Intelligence and Knowledge Engineering for Robotic Applications

Edited by

Andrew K. C. Wong

Systems Design Engineering, University of Waterloo
Waterloo, Ontario N2L 3G1, Canada

Alan Pugh

Electronic Engineering, University of Hull
Hull HU6 7RX, United Kingdom

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo
Published in cooperation with NATO Scientific Affairs Divison

Proceedings of the NATO Advanced Research Workshop on Machine Intelligence and Knowledge Engineering for Robotic Applications held at Maratea, Italy, May 12–16, 1986

ISBN 3-540-17844-9 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-17844-9 Springer-Verlag New York Berlin Heidelberg

Library of Congress Cataloging in Publication Data. NATO Advanced Research Workshop on Machine Intelligence and Knowledge Engineering for Robotic Applications (1986 : Maratea, Italy) Machine intelligence and knowledge engineering for robotic applications. (NATO ASI series. Series F, Computer and systems sciences ; vol. 33) "Published in cooperation with NATO Scientific Affairs Division." "Proceedings of the NATO Advanced Research Workshop on Machine Intelligence and Knowledge Engineering for Robotic Applications held at Maratea, Italy, May 12–16, 1986."—T.p. verso. Includes index. 1. Robotics—Congresses. 2. Artificial intelligence—Congresses. I. Wong, Andrew K.C. II. North Atlantic Treaty Organization. Scientific Affairs Division. III. Title. IV. Series: NATO ASI series. Series F, Computer and systems sciences ; vol. 33. TJ210.3.N375 1986 629.8'92 87-12863
ISBN 0-387-17844-9 (U.S.)

Printed in Germany

Printing: Druckhaus Beltz, Hemsbach; Bookbinding: J. Schäffer GmbH & Co. KG, Grünstadt
2145/3140-543210

ABOUT THE BOOK

This book consists of a collection of papers addressing the topic: "Machine Intelligence and Knowledge Engineering for Robotic Applications". It is mainly an outcome of a NATO Advanced Research Workshop held at Maratea, Italy in May 1986. The authors are internationally known and hence the book may be considered an important documentation of the state of the art in knowledge-based robotics.

The book presents and reviews the recent advances in this significant field of research. It covers: robot vision, knowledge representation and image understanding, robot control and inference systems, task planning and expert systems and integrated software and hardware systems. Each of these areas is addressed by several authors who approached the problems differently. Almost all the articles attempt not only to consider the theoretical aspects, but also to include in their presentation challenging issues such as systems implementation and industrial applications.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the professional support and financial assistance provided by the late Dr. Mario di Lullo and the Scientific Affairs Division, NATO, Brussels, Belgium. Professors E. Baker, L. Pau, A. Pugh, A. Sanderson and A.K.C. Wong, members of the Organization Committee of the Machine Intelligence and Knowledge Engineering for Robotic Applications Advanced Research Workshop also deserve special gratitutde. Without their dedicated support, the Workshop could not have achieved what it did. The Editors recognize, too, the contribution of Anne Ross and Deborah Stacey, whose tireless effort made this publication possible.

PREFACE

This book is the outcome of the NATO Advanced Research Workshop on Machine Intelligence and Knowledge Engineering for Robotic Applications held at Maratea, Italy in May 1986. Attendance of the workshop was by invitation only. Most of the participants and speakers are recognized leaders in the field, representing industry, government and academic community worldwide.

The focus of the workshop was to review the recent advances of machine intelligence and knowledge engineering for robotic applications. It covers five main areas of interest. They are grouped into five sections:

1. Robot Vision
2. Knowledge Representation and Image Understanding
3. Robot Control and Inference Systems
4. Task Planning and Expert Systems
5. Software/Hardware Systems

Also included in this book are a paper from the Poster Session and a brief report of the panel discussion on the Future Direction in Knowledge-Based Robotics.

Section I of this book consists of four papers. It begins with a review of the basic concepts of computer vision, with emphasis on techniques specific for robot vision systems. The next paper presents a comprehensive 3-D vision system for robotic application. It covers various theoretical aspects and factory applications of 3-D robot vision. The last two papers deal with computer vision on moving objects. The first provides an overview on the computation of motion from a sequence of monocular or stereo images. The second describes new techniques on time-varying image analysis with a high-level system for representing and identifying time-varying characteristics of a large class of physical events.

Section II is on Knowledge Representation and Image Understanding. The first paper presents a general and flexible knowledge representation using attributed graphs and hypergraphs as the basic data structure. With these representations, model synthesis and recognition of 3-D objects based on various forms of graph morphism algorithms are described. Also presented is a knowledge directed method for recognizing and locating 3-D objects from a single 2-D perspective image. An attempt is made to show that similar representation can be used to represent the world environment of a roving robot in both path planning and navigation. As for the second paper, it describes a knowledge-based system for robotic applications. After a general discussion on knowledge-based system which is able to acquire knowledge in specified domains, store knowledge in defined structure and organize knowledge in desired format for access, retrieval, transfer, utilization and extension, it presents some recent results on a piloted vision system for roving robot navigation, label reading and 3-D object recognition. The next paper addresses image understanding for robotic application. It describes how knowledge about the robot environment could be used by an image understanding algorithm to facilitate the recovery of information from images. Two specific applications are described in the paper: one demonstrates how knowledge of ego-motion parameters of a mobile robot could be used for segmentation of a scene and the recovery of depth information, and the other shows how a hypothesize-and-test approach could be used to find road edges in real scenes for an autonomous vehicle.

Section III covers the use of machine intelligence and knowledge-based systems for robot control and target tracking. The first paper presents a hierarchical control approach for machine intelligent robots. It is based on what is defined as Hierarchical Intelligent Control and the Principle of Decreasing Precision with Increasing Intelligence. Entropy is used as a common measure for the probalistic model involved. The second paper is a critique on the application of artificial intelligence planning technique in industrial robots. It shows the limitation of some classical A.I. paradigms in industrial application and recommends directions for future development. The third paper investigates some fairly universal concepts of analogical reasoning in the context of the block world. Frames are used to represent problem situations and the three-stage underlying learning process is also described. The fourth paper is on the overall hardware and software architecture of a knowledge-based system for change detection, target tracking and threat assessment. Based on target features, numbers and maneuver patterns or changes in the scene, the system is able to assign threat level and threat scenario labels to the scene. Thus interpretation of the scene could be achieved more efficiently and reliably.

Section IV is on task planning and expert systems. The first paper covers the conceptual, the algorithmic and data structure of a task and path planning project for a mobile robot. It presents a world model which includes: a) a set of elementary task operators; b) three-level environment models, namely, geometrical, topological and semantic, and c) the functional capabilities of the machine in the form of specialized processing modules.

The next paper describes a nonlinear planning approach for task planning and control synthesis for flexible assembly systems. It proposes assembly sequences based on relational model of part contacts and attachments. The resulting plans are then consolidated into AND/OR graph representation which provides a basis for efficient scheduling of operations. A simple example is used to demonstrate the efficiency of this approach in comparison to a fixed sequence method. The last paper of this section presents a robust and practical expert robot welding system called MARS. It identifies the various relevant variables of the welding process and investigates their interrelation so as to develop a mathematical model for feedback control of a welding robot. The objective of the project is to construct a computerized hierarchical expert welder.

Section V consists of three papers describing several software/hardware robot systems. The first is on the Edinburgh Designer System which can serve as a general framework to support symbolic computing for robotics. It concludes that a) an algebra engine is required to handle temporal constructs, groups and tolerances; b) a proposed taxonomy can support activity modules, c) and an automatic plan formation would require the creation of a "specialist". The second paper describes the implementation of complex robot subsystems through distributing computational load functionally over several micro-processor systems in both tightly and loosely coupled configurations. This approach is used to explore various concepts of sensor data fusion. An autonomous mobile robot which has provided the experimental environment is also described. The third paper discusses the autonomous research robot being developed at the University of Karlsruhe. The device is able to perform simple operations in the laboratory. It contains a mobile platform, a complex sensor system, two manipulators, hierarchical controls and an expert system. The paper describes how the fundamental technology developed at the Institute is being integrated in the robot system. The last paper summarizes the research and development activities in the field of intelligent robotics at the Laboratory for Intelligence Systems for the National Research Council of Canada. It gives a brief description on the Council's objective and introduces several projects of the research laboratory on 3-D vision, sensory based control, multi-processor system architecture and applications of artificial intelligence. The last paper of this book is a closing remark based on the Panel Discussion on the Future Direction in Knowledge-Based Robotics. It summarizes the general discussions, recommendations and future directions of each of the areas covered under the five sections. The Panel Discussion concluded on an optimistic note, with researchers targeting sensor fusion, advance computer architecture and an increase in intelligence in all systems (knowledge base and sensor) as areas to be actively pursued.

A.K.C. Wong

TABLE OF CONTENTS

ROBOT VISION

Azriel Rosenfeld
Center for Automation Research
University of Maryland
College Park, MD 20742/USA

1. INTRODUCTION

This article reviews the basic concepts of computer vision, with emphasis on techniques that have been used, or could be used, in robot vision systems. Sections 2 and 3 discuss two- and three-dimensional vision systems, respectively, while Section 4 briefly discusses some other vision topics. References to basic papers or review papers are given in connection with each topic.

2. TWO-DIMENSIONAL VISION

The general goal of computer vision is to derive a description of a scene by analyzing one or more images of the scene. In many situations the scene itself is basically two-dimensional; for example, a robot might be dealing with flat parts lying on a flat surface, or might be looking for holes in such a surface. Vision is much easier for two-dimensional (2D) scenes, and not surprisingly, the earliest work on vision dealt with such scenes. This section outlines the basic steps in the 2D vision process, and then presents a review of some of the methods used to carry out these steps. Three-dimensional vision is more complicated; it will be treated in the next section.

2.1. The 2D Vision Process

In order for a robot to recognize parts, holes, etc. on a surface—in general: "objects"—it must first be able to distinguish the objects form the rest of the surface. In other words, it must be able to single out pieces of the image that (hopefully) correspond to the objects. This process of extracting subsets of an image that correspond to relevant parts of the scene is called *segmentation*.

Once a subset has been extracted from an image, it is usually necessary to measure various geometric properties of the subset (size, shape, etc.). These measurements can serve as a basis for recognizing the subset as representing a given object, for determining the position and orientation of the object, etc. They may also serve as a basis for further segmenting the subset; for example, if two objects touch or overlap, they may

NATO ASI Series, Vol. F33
Machine Intelligence and Knowledge Engineering for Robotic Applications. Edited by A.K.C. Wong and A. Pugh

be extracted as a single subset, and it may than be necessary to break the subset into two pieces on the basis of geometric criteria (e.g., to break it into convex pieces). This stage of the computer vision process is called *geometric analysis*. Different algorithms for geometric analysis can be designed, depending on how the image subsets are represented in the computer; thus a topic closely related to geometric analysis is *geometric representation* of image subsets.

Recognition of objects by analyzing subsets of an image can vary greatly in difficulty, depending on the complexity of the objects. If the objects that might be present in the scene differ greatly from one another, simple "template matching" techniques can be used to distinguish them; in this situation it may not even be necessary to explicitly extract the objects from the rest of the image. More generally, objects can often be recognized by the fact that they have a characteristic set of geometric property values; this "feature matching" approach was used in the well-known "SRI vision module" (1). If the objects are complex, it may be necessary to break up the recognition process into stages: to first detect subobjects and recognize their properties, and then to recognize the objects as combinations of subobjects in specific relationships; this is known as the "structure matching" approach.

2.2. Segmentation

A *digital image* is a discrete array of numbers representing brightness values at regularly spaced points in the scene. The elements of a digital image are called *pixels*, and their values are called *gray levels*. (Color has not been extensively used as yet in robot vision systems; color images will be briefly discussed in the next subsection.) This section reviews basic methods of segmenting digital images into subsets. For general surveys of image segmentation see (2,3). The effectiveness of a segmentation technique depends on the properties of the class of images to which it is applied (4); approaches to defining, or "modeling", classes of images are reviewed in (5).

2.2.1. Thresholding

If an object differs significantly in brightness from its background, it gives rise to a set of pixels in the image that have significantly different gray levels from the rest of the image. (Large brightness differences between an object and its background can often be produced by controlling the illumination so as to silhouette or edge-light the object.) Such image subsets can be extracted from the image by *thresholding* the pixel gray levels, e.g., classifying a pixel as "light" or "dark" depending on whether its gray level lies above or below a specified "threshold" level.

If the illumination of the scene can be controlled and the sensor can be calibrated, it may be possible to set a threshold once and for all to correctly segment scenes of a given class; but in general it will be necessary to determine the best threshold for each individual image. If the objects occupy a significant fraction of the scene, this can be done by examining the *histogram* of the image, which is a graph showing how often each gray level occurs in the image (6). This histogram should have two peaks, one representing background gray levels and the other object gray levels; these ranges of gray levels give rise to peaks because they occur relatively frequently. Intermediate gray levels should be relatively rare, and should give rise to a valley on the histogram, between the peaks. A good gray level at which to set the threshold is evidently the level corresponding to the bottom of the valley, since nearly all object pixels and nearly all background pixels will be on opposite sides of this threshold.

If the illumination of the scene is not uniform, dark objects at one side of the scene may actually be brighter than the light background at the other side, so that the objects cannot be separated from the background by simple thresholding. One way to handle this situation (7) is to divide the image into blocks and pick a threshold for each block by analyzing its histogram. These thresholds can then be interpolated to yield a "variable threshold" that properly segments the entire image. A survey of threshold selection techniques can be found in (8).

The color at a point in a scene can be characterized by a triple of numbers representing, for example, the values of red, green, and blue "color components". Thus a digital color image is an array of triples of values. If these pixel values are plotted as dots in "color space", an object (or background) of a given color gives rise to a cluster of dots. These clusters are analogous to histogram peaks, and the image can be segmented into regions having different colors by partitioning the color space so as to separate the clusters. This approach is classically used to segment images obtained by multispectral scanners in remote sensing, but it has not yet found significant use in robot vision.

2.2.2. Edge detection

Small objects are not easy to extract from their background by thresholding, because the histogram peaks that they produce may be too small to detect reliably. Similarly, if a scene contains many objects of different brightnesses, it is not easy to extract them by thresholding, because their histogram peaks overlap. Another method of segmentation can be used in such cases, provided the objects have relatively uniform brightness and that they contrast strongly with their immediate backgrounds. This implies that the rate of change of gray level is low within the objects, but high at the borders of the objects. The objects can thus be extracted by *edge detection*, i.e., by detecting pixels at which the rate of change of gray level is high.

The classical method of detecting edges in an image is to apply an isotropic derivative operator, such as the gradient operator, to the image; such an operator will have high values at edges, no matter what their orientations (9). Many digital approximations to the gradient have been used for this purpose; an especially simple example is the "Roberts cross" operator (10), and another frequently used operator is the "Sobel operator" (11).

Several other basic methods of edge detection are the following: (a) Match the image in the vicinity of each pixel with "templates" of step functions in different orientations; if a good match is detected, an edge in that orientation is likely to be present (12). (b) Fit a polynomial surface to the image gray levels in the neighborhood of each pixel; if the gradient of the fitted surface is high, an edge is likely to be present (12). (c) Fit a step function to the image gray levels in the neighborhood of each pixel; if this step has high contrast, an edge is likely to be present (13). (d) Apply a Laplacian operator to the image; the zero-crossings of the Laplacian values correspond to edges (14). An early survey of edge detection techniques can be found in (15).

2.2.3. Texture analysis

If an object is not uniform in brightness, but rather is patterned, neither thresholding nor edge detection can be used to extract it, since its pixels do not have gray levels in a narrow range, and it has many internal edges. Nevertheless, such an object may be distinguishable from its background because of its characteristic pattern of gray levels, or "visual texture". For a general survey of visual texture analysis see (16).

Textures can be characterized by sets of local properties of their pixels, i.e., by the fact that in a textured region, certain local patterns of gray levels tend to be present in the neighborhood of each pixel. An early survey of local properties that can be used to distinguish textures can be found in (17). By computing a set of such properties at each pixel, the pixel can be characterized by a set of numbers (compare the discussion of color images in the subsection on thresholding), and the image can be segmented into differently textured regions by partitioning the "local property space" so as to separate the clusters corresponding to the regions. Since local properties tend to be more variable than colors, some degree of local averaging should be performed first in order to make the clusters more compact. Similarly, by computing average values of local properties and then taking differences of these averages, one can compute a "texture gradient" at each pixel and use it to detect "texture edges", i.e., edges between differently textured regions (18).

A powerful method of characterizing textures is by performing various types of "shrinking" and "expanding" operations on them and analyzing the results; for example, thin patterns disappear under small amounts of shrinking, while closely spaced patterns "fuse" under small amounts of expanding. This approach to image analysis has been used in a variety of applications for over 20 years; a recent comprehensive treatment is (19).

2.2.4. Tracking and region growing

The methods of segmentation discussed so far treat each pixel (or its neighborhood) independently; they are oblivious as to whether the resulting pixels constitute a connected region, or whether the resulting edge segments constitute a smooth, high-contrast boundary. Better-quality regions or edges can be obtained by requiring that the results be locally consistent, i.e., that the regions be connected or that the edges smoothly continue one another. Methods of "tracking" edges sequentially, pixel by pixel, or of "growing" regions, can be used to insure continuity. (A survey of region growing techniques can be found in (20).) A more powerful, but computationally more expensive, approach is to require (piecewise) global consistency, e.g., to search for regions that are optimal with respect to constancy or smoothness of gray level, or for edges that are optimal with respect to contrast and smoothness of direction. A useful approach to finding globally consistent regions is a split-and-merge process in which regions are split if they are inconsistent, and pairs of adjacent regions are merged if their union is consistent. For a general treatment of image segmentation by partitioning into consistent regions see (21).

2.3. Geometric Analysis

Once a region has been segmented from an image, it can be represented by a "binary image" in which pixels belonging to the region have value 1, and those belonging to the background have value 0. Various geometric properties of the region can be computed from this binary image at low computational cost. This process is sometimes referred to as *binary vision*. The following subsections discuss basic geometric properties and their measurement, as well as other, more compact ways of representing regions.

2.3.1. Connectivity and borders

If a scene contains several objects on a background, segmenting an image of the scene yields the entire set of pixels belonging to all the objects; it does not distinguish the objects from one another. In order to deal with one object at a time, it is necessary to "label" the object pixels so that pixels belonging to the same object get the same label (and conversely). This process is called *connected component labeling* (22); it assigns distinctive labels to sets of object pixels that are all mutually connected. The theory of connected regions in digital pictures is developed in (23). Connectedness is the basic principle that underlies the process of counting objects (i.e., connected regions) in an image.

The *border* of a region consists of these region pixels that are adjacent to non-region pixels. These border pixels lie on a set of curves, one representing the "outer border" of the region and the others representing the borders of its holes, if any. To label these borders individually, a "border following" process can be used (22) that, starting from any border pixel, successively visits all the pixels belonging to that border until it returns to the starting pixel.

2.3.2. Size and shape properties

The *area* of a region is (approximately) the number of it pixels (i.e., the number of pixels having a particular component label). The *perimeter* is the number of border pixels, or (for a specific border) the number of moves required to follow the border completely around. A frequently used shape measure is area/(perimeter)^2, which measures the *compactness* of the region. The *elongatedness* of a region can be defined using a process of shrinking and area measurement; a region is elongated if it has large area but disappears under a small amount of shrinking. *Distance* measures are another source of useful shape information; on measures of distance in a digital image see (24), and on the approximation of Euclidean distance see (25).

Many shape properties of a region can be derived by measuring the *curvature* (i.e., rate of change of direction) of its border. *Concavities* correspond to parts of the border where the curvature is "negative" (in the sense that direction is changing counterclockwise while the border is being followed clockwise). (On the theory of concavity in digital images see (26); on the characterization of straight line segments see (27).) *Corners* are border points where the curvature has a high (positive or negative) value. Such properties are useful in segmenting a region into parts when necessary; for example, when two objects in the scene touch or overlap, they give rise to a single connected region in the image, but they can be "cut apart", e.g., by making a cut that joins the bottoms of two deep concavities. A review of shape analysis algorithms for contours (i.e., borders) can be found in (28), and a general survey of shape analysis techniques can be found in (29).

The *moments* of a region provide useful information about its shape (30). The (i, j) moment m_{ij} is defined as $\sum x^i y^j$ summed over all pixels (x, y) of the object. (Moments can also be defined for gray-level images by weighing pixel (x, y) by its gray level.) Thus m_{00} is the area of the region, and $(m_{10}/m_{00}, m_{01}/m_{00})$ are the coordinates of the *centroid* of the region. The *principal axis* of a region is the line (through the centroid) about which the region's moment of inertia is least; its slope $\tan\theta$ satisfies the quadratic equation $\tan^2\theta + (m_{20} - m_{02})\tan\theta/m_{11} - 1 = 0$, where the m's are moments