FAST 2D SHAPE APPROXIMATION

Algorithms And Their Errors

Dipl.-Ing. W E Friedrich

A thesis submitted in fulfillment of the requirements for the
degree of Doctor of Philosophy in Engineering
University of Auckland,
New Zealand, 1989.

ABSTRACT

In this thesis, efficient algorithms for fast shape approximation in the area of machine and robot vision are analysed. The core of the thesis consits of two main parts, an **error analysis** of chain coded silhouettes and a simplified, hence fast **shape approximation** scheme for practical applications.

The area of shape description and approximation is introduced. General techniques for processing of boundary data are outlined followed by a brief description of the image processing system which was developed during the study. Processing procedures developed and their options are presented. Two applications realized by the author have emerged from the algorithm development, firstly a semi-automatic shape registration and measurement procedure and secondly, a low-cost robot vision system. The advantage of the experimental robot vision system is an effective communication between the robot and the object recognition system.

The shortcomings of these applications, mainly inaccuracy of the shape description lead to the introduction of an error analysis for boundary descriptors. In contrast to existing methods, the **error analysis** can be applied to straight lines, circular arcs, and arbitrary shapes consisting of these two shape primitives. The analysis is comprehensive and covers all kinds of chain code sets representing various pixel shapes as well as different pixel configurations.

The results of the analysis lead to the development of simple ways of overcoming the accuracy limitations of conventional methods. Two algorithms for improving the length estimation of object outlines are introduced. A comparison with existing methods showing the effectiveness of these algorithms is made.

For practical applications of object recognition, a new, simple and hence fast and effective algorithm for polygon approximation, called the "**arc operator**" is developed. The combination of the arc operator and the length correcting algorithm directly improves the ability to identify partially visible shapes. The performance of the arc operator is compared to six other well known algorithms. Slight disadvantages in accuracy are outweighed by the enormous advantage in processing time, particularly with an integer implementation. An example of overlapping parts demonstrates the potential of the arc operator to extract characteristic shape descriptions.

ACKNOWLEDGEMENTS

I am indebted to the Department of Scientific and Industrial Research (DSIR) for providing the funding for this study. I would also like to thank the following people who have assisted me during the course of this study.

Associate Professor Gunter Arndt and Dr. Des Tedford
of the Mechanical Engineering Department
for their part as supervisors, their advice and encouragement.

Wayne Power for proof reading,

Mike Riley for expert advice on page layout, and last but not least

Denis Malone and John de Pont for helpful discussions.

I am very grateful to my wife Ute for her understanding and moral support.

To her and our son Stefen I dedicate this thesis.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Symbols

A, B and C = Error function coefficients
a = half the side length of regular geometric shapes considered
d = arc operator distance and its x and y components (index)
d_s = boundary element length
d_x = position difference in x direction
d_y = position difference in y direction
Err = error function (length deviation error)
(index indicates code set, e.g. Err_4 for 4 direction code
v for vertical
d for diagonal
h for horizontal
ave = average
tot = total
a, b, c, d, e, f, g, h, i, k, l, p, q, r, s =
= are used as parameters to calculate the error matrix
f_c = Digitization / conversion frequency
f_s = image frame rate per second
i = pixel index
k = pixel neighbourhood size, and arc length
k = arc operator length
L_e = estimated length of a digitized line
l_f = lines per frame
l_i = individual length of code element
L_t = true length
m = number of equal sides of a polygon
n_i = frequency of code element
r_c = aspect ratio of camera
r = pixel aspect ratio
α = slope angle of straight line or shape segment
β = orientation of shape
ϕ = code set angle
v_p = vector descriptor
x(i), y(i) = x and y coordinates of the border pixel at the i^{th} location in the code set or boundary.