



<http://researchspace.auckland.ac.nz>

*ResearchSpace@Auckland*

### **Copyright Statement**

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage.

<http://researchspace.auckland.ac.nz/feedback>

### **General copyright and disclaimer**

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form.

# **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.

Thesis

1989-F91

Copy 1

90/6/954886/0/01

14/9/90

## 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 consists 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

---

<b>ABSTRACT</b>	i
<b>ACKNOWLEDGEMENT</b>	ii
<b>List of Figures</b>	vi
<b>List of Tables</b>	viii
<b>List of Symbols</b>	ix
 <b>1. INTRODUCTION</b>	 1.1
 <b>2. SHAPE DESCRIPTION ALGORITHMS</b>	
2.1 Basics of Machine Vision	2.1
2.1.1 Processing stages of machine vision	2.1
2.1.2 Relation to adjacent fields	2.3
2.2 Digital Image Analysis	2.4
2.2.1 Segmentation	2.4
2.2.2 Shape description	2.5
2.2.3 Global parameters	2.5
2.2.4 Local descriptors	2.6
2.2.5 Chain codes and Crack codes	2.7
2.2.6 Object recognition	2.8
2.3 Industrial Vision Systems	2.9
2.4 Laboratory Equipment & Development	2.10
2.4.1 Equipment	2.11
2.4.2 Kiwi fruit measurement	2.13
2.4.3 Robot vision	2.15
2.4.4 Shortcomings of the two systems	2.19
2.5 Shape of picture elements	2.19
2.5.1 Quantization	2.19
2.5.2 Square pixels	2.20
2.5.3 Rectangular pixels	2.20
2.5.4 Hexagonal pixels	2.20
2.6 Algorithms to Improve Chain Code Descriptions	2.21
2.7 Shape Approximation	2.22
2.8 Discussion	2.24

### 3. THEORETICAL ERROR ANALYSIS FOR CHAIN CODED STRAIGHT LINES

3.1 Introduction	3.1
3.1.1 Pixel shapes and configurations	3.1
3.1.2 Chain Code sets for different pixel configurations	3.2
3.2 Estimating The Length of Digitized Straight Lines	3.3
3.2.1 Crack code errors	3.4
3.2.2 Chain code errors	3.6
3.2.3 Hexagonal code errors	3.7
3.3 The Fundamental Error Function	3.8
3.3.1 Slope ranges for square and rectangular pixels	3.10
3.3.2 Slope ranges for hexagonal pixels	3.11
3.4 Maximum and Average Magnitudes of Length Deviation	3.11
3.4.1 Square pixel error function	3.12
3.4.2 Rectangular pixel error function	3.13
3.4.3 Hexagonal pixel error function	3.14
3.4.4 Extended code sets	3.14
3.5 The Matrix Form of The Error Function	3.21
3.6 Conclusion: Error Analysis	3.25

### 4. PERIMETER ESTIMATION ERROR OF CHAIN CODED SILHOUETTES

4.1 Introduction	4.1
4.2 Principle of Length Deviation Using the Error Function	4.1
4.3 The Correlation of the Error Function	4.5
4.4 Examples of Vector Representations for Simple Shapes	4.6
4.4.1 Regular polygons	4.6
4.4.2 Arbitrary polygons	4.6
4.4.3 Circular arcs	4.7
4.4.4 Combined circular arcs and straight line segments	4.8
4.5 Examples of Specific Error Profiles	4.8
4.6 Conclusion	4.10

### 5. SILHOUETTE APPROXIMATION BY CHAIN CODES

5.1 Introduction	5.1
5.2 Description of Silhouette Coding Methods	5.3
5.2.1 Crack code methods	5.3
5.2.2 Chain code methods	5.6
5.3 Experiments with Simple Shapes	5.8
5.3.1 Silhouette generation	5.8
5.3.2 Code Generation	5.9
5.3.2.1 Chain code generation	5.9
5.3.2.2 Crack code generation	5.10
5.3.3 Experiments with 4 chain coded shapes	5.11
5.3.3.1 Circles	5.11
5.3.3.2 Semi-circles	5.13
5.3.3.3 Squares and triangles	5.15
5.4 Conclusion	5.18

## 6. THE ARC OPERATOR

6.1 Introduction	6.1
6.2 A New Approach to Shape Analysis	6.2
6.2.1 Definition of the algorithm	6.2
6.2.2 Distance function	6.4
6.2.3 Running distance total	6.5
6.3 Parameter Selection	6.8
6.3.1 Segment length	6.8
6.3.2 Perpendicular distance	6.8
6.3.3 Integrated distance	6.9
6.3.4 Segment deviation	6.10
6.4 Implementation Considerations	6.12
6.5 Experimental Results	6.15
6.6 The Slope Angle Domain Application	6.23
6.6.1 Feature recognition	6.23
6.6.2 Slope angle approximation	6.25
6.6.3 Overlapping objects	6.27
6.7 Concluding Remarks	6.30

## 7. CONCLUSION AND FUTURE WORK

7.1

## 8. REFERENCES

8.1

## APPENDIX

Overview of appendix organization	A.1
<b>A</b> Error Function - Coefficients and Matrices.	A.2
<b>B</b> Catalogue of Error Function Profiles	B.1
<b>C</b> Software Library - Implemented Image Processing Functions	C.1
<b>D</b> Robot Sensor Communication and Protocol	D.1
<b>E</b> Publications	
Vision-Aided Flexible Component Handling, Robhanz, Nov 1986 also in <b>Automation and Control (NZ)</b> , 17(10):16-21, 1986.	E.1
A New Silhouette-Related Algorithm for Improving Vision-Aided Robot Applications, <b>Annals of the CIRP</b> Vol. 37 / 1 / 1988, p 481 - 484.	E.5
Integrated Robot Vision System for Flexible Component Handling <b>SPIE Conference on Visual Communications and Image Processing '89</b> , Philadelphia, USA.	E.9



## List of Figures

---

Figure 2-1:	Image analysis by machine vision	2. 2
Figure 2-2:	Relation to adjacent fields based on Pavlidis.	2. 3
Figure 2-3:	Chain code and Crack code definition.	2. 7
Figure 2-4:	Hierarchy of program library and structure.	2. 13
Figure 2-5:	Kiwifruit work station	2. 14
Figure 2-6:	Average Kiwifruit shape	2. 15
Figure 2-7:	Robot Vision set-up at Auckland University	2. 16
Figure 2-8:	Rapid object position test	2. 17
Figure 2-9:	Schematics of the robot vision system	2. 18
Figure 3-1:	Pixel shapes and configurations	3. 2
Figure 3-2:	Chain code sets for $k = 1, 2$ and $3$	3. 3
Figure 3-3:	Code sets for hexagonal configurations	3. 3
Figure 3-4a:	Error definition for the crack codes	3. 4
Figure 3-4b:	Error function for a 4 direction code	3. 4
Figure 3-5a:	Error definition for the 8 direction code	3. 6
Figure 3-5b:	Error function for an 8 direction code	3. 7
Figure 3-6:	Error function for hexagonal pixels ( $k = 1$ )	3. 8
Figure 3-7:	Approximation triangle	3. 8
Figure 3-8a:	Error function for rectangular pixels	3. 13
Figure 3-8:	Error function for digitized straight lines	3. 14
Figure 3-9:	Average error for digitized straight lines	3. 16
Figure 3-10:	Error function for 16 direction code	3. 16
Figure 3-11:	Error function for 24 direction code	3. 17
Figure 3-12:	Error function for hexagonal pixels	3. 18
Figure 3-13a:	Error maxima for 72 direction codes	3. 18
Figure 3-13b:	Error maxima for triplet and hexagonal pixels	3. 19
Figure 3-14:	Angle intervals for $k = 9$ and $k = 12$	3. 19
Figure 3-15:	Average errors for $k = 1$ (8 dir) to $k = 100$ (800 dir)	3. 20
Figure 3-16:	Flow chart for error matrix values	3. 23
Figure 3-17:	Matrix values compared to error function	3. 24
Figure 4-1:	Deviation error of a square	4. 2
Figure 4-2:	Deviation error of a rectangle	4. 3
Figure 4-3:	Error values of a square at 60 degrees	4. 4
Figure 4-4:	Error profile of a square (rect. pixels)	4. 4
Figure 4-5:	Vector diagram of a regular triangle	4. 6
Figure 4-6:	Vector diagram of a rectangle (ratio 2:1)	4. 7
Figure 4-7:	Error profile of a semi-circle	4. 9
Figure 4-8:	Error profile of a triangle	4. 10

Figure 5-1:	Arc approximation with various code sets	5.2
Figure 5-2:	Error function for a corrected square	5.3
Figure 5-3:	Silhouette approximation by Dunkelberger	5.4
Figure 5-4:	Silhouette approximation by Friedrich	5.5
Figure 5-5:	Fish-bone pattern for code detection	5.9
Figure 5-6:	Error comparison for chain coded circles	5.12
Figure 5-7:	Error comparison for crack coded circles	5.12
Figure 5-8:	Error comparison for crack coded semi circles	5.13
Figure 5-9:	Error comparison for chain coded semi circles	5.14
Figure 5-10:	Error comparison for crack coded triangles	5.15
Figure 5-11:	Error comparison for chain coded triangles	5.16
Figure 5-12:	Error comparison for crack coded squares	5.16
Figure 5-13:	Error comparison for chain coded squares	5.17
Figure 5-14:	Results of methods examined	5.18
Figure 6-1:	Principle of the arc operator	6.3
Figure 6-2:	Corner detection with the arc operator	6.5
Figure 6-3:	Reconstructed outline of metal stamping	6.6
Figure 6-4:	Curve detection using a Kiwi fruit silhouette.	6.7
Figure 6-5:	Polygon approximation of a Kiwi fruit.	6.7
Figure 6-6:	Influence of a large region of support.	6.8
Figure 6-7:	Distance threshold only.	6.9
Figure 6-8:	Integrated distance threshold.	6.10
Figure 6-9:	Segment distance.	6.11
Figure 6-10:	Polygon approximation for a seat belt buckle.	6.11
Figure 6-11:	Simplified distance measure	6.12
Figure 6-12:	Possible hardware implementation of the Arc operator.	6.13
Figure 6-13:	Arc operator functions on a sharp corner sample.	6.14
Figure 6-14:	Competitive break-point reporting	6.15
Figure 6-15:	Split and merge method [Pavlidis 1982]	6.16
Figure 6-16:	Arc operator with floating point calculations.	6.17
Figure 6-17:	Arc operator with integer calculations.	6.17
Figure 6-18:	Split and merge approximation (circular stamping)	6.19
Figure 6-19:	Arc operator approximation (circular stamping).	6.19
Figure 6-20:	Split and merge approximation (screw terminal).	6.20
Figure 6-21:	Arc operator scheme applied to a screw terminal	6.20
Figure 6-22:	Split and merge approximation (chuck key).	6.21
Figure 6-23:	Arc operator approximation (chuck key).	6.21
Figure 6-24:	Slope angle diagram of the switch cover - straight outlines	6.24
Figure 6-25:	Semi-circular object [Teh 1988].	6.24
Figure 6-26:	Slope angle representation of seat belt buckle	6.24
Figure 6-27:	Approximating the slope angle profile (semi-circular shape)	6.26
Figure 6-28:	Approximating the slope angle profile (circular stamping)	6.26
Figure 6-29:	Overlapping objects	6.27
Figure 6-30:	Slope diagram of overlapping objects	6.28
Figure 6-31:	Slope diagram of circular stamping (conventional approx)	6.28
Figure 6-32:	Approximated slope angle diagram of chuck key.	6.29
Figure 6-33:	Approximated slope angle diagram of screw terminal.	6.29
Figure 6-34:	Matching trial in the slope angle domain.	6.30

## List of Tables

---

Table 3-1:	Length estimation errors for chain codes.	3 . 15
Table 3-2:	Error matrix for short digitized straight lines	3 . 24
Table 5-1:	Code combinations and their length replacement factors	5 . 6
Table 5-2:	Code approximation replacement table [Dessimoz]	5 . 7
Table 5-3:	Organization of a lookup table for length adjustments	5 . 7
Table 5-4:	8 direction chain code from position differences	5 . 10
Table 5-5:	Chain code conversion	5 . 11
Table 6-1:	Performance comparison of the arc operator with various other algorithms from [Teh 1988].	6 . 18
Table 6-2:	Performance comparison of the arc operator with the split and merge algorithm [Pavlidis 1982].	6 . 22

## 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. Err4 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

$\alpha$  = slope angle of straight line or shape segment

$\beta$  = orientation of shape

$\phi$  = code set angle

$v_p$  = vector descriptor

$x(i), y(i)$  = x and y coordinates of the border pixel at the  $i^{\text{th}}$  location  
in the code set or boundary.