Investigation of the process in the construction of 3D models from 2D sketches

Sun, Yuan

2006

Sun, Y. (2006). Investigation of the process in the construction of 3D models from 2D sketches. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/6418>

<https://doi.org/10.32657/10356/6418>

Nanyang Technological University

Downloaded on 26 Apr 2023 21:12:33 SGT

INVESTIGATION OF THE PROCESS IN THE CONSTRUCTION OF 3D MODELS FROM 2D SKETCHES

SUN YUAN

SCHOOL OF MECHANICAL & AEROSPACE ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

INVESTIGATION OF THE PROCESS IN THE CONSTRUCTION OF 3D MODELS FROM 2D SKETCHES

Sun Y. 2006

2006

Investigation of the Processes in the Construction of 3D Models from 2D Sketches

Sun Yuan

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2006

Acknowledgements

I wish to express my sincere gratitude and appreciation to my supervisor, A/Prof. Lee Yong Tsui for his invaluable advice and guidance.

My sincere appreciation is also extended to A/Prof. Chen Chen Hsien, A/Prof. Au Chu Kit and A/Prof. Lin Feng, for their enlightening comments in my first year report.

I am really appreciated Dr. Liu Jianzhuang, Dr. Sun Jie, Mr. Wu Tao and Ms. Zou Hongliu. They give me valuable help in my research.

The technicians in Design Research Center are also greatly acknowledged here.

I also thank my friends. Their friendship helped me out in many ways and made the life here warm and colorful.

I would like to send my utmost thanks and appreciation to my family. Their support always gives me extra power to adventure ahead.

Finally, I would like to dedicate this thesis to my mother and grandmother.

CONTENTS

Acknowledgements	i
Table of Contents	ii
List of Figures	ix
List of Tables	xi
List of Symbols	xii
Summary	xviii

Table of Contents

Chapter 1 Introduction	1
1.1 The importance of sketching	1
1.2 Sketching in conceptual design and CAD system	3
1.2.1 Sketching in conceptual design	3
1.2.2 Sketching with CAD	5
1.3 Literature review	7
1.3.1 An overview of sketching-based system	8
1.3.2 Gesture-based sketching system	9

1.3.3 Line drawing-based interpretation and reconstruction system	11
1.3.3.1 Line labeling approach	11
1.3.3.2 Optimization-based reconstruction methods	13
1.3.3.3 3D reconstruction systems applying the hybrid technique	20
1.4 Research objectives and contributions	21
1.5 Organization of the thesis	23
Chapter 2 Optimization-based 3D Reconstruction Method from 2D Sketches	26
2.1 Information processing scheme in 3D reconstruction system	26
2.2 Regularities for 3D reconstruction	28
2.3 Optimization-based 3D reconstruction method	32
2.3.1 Optimization process for 3D reconstruction	32
2.3.2 Compliance function for the 3D reconstruction	33
2.3.3 Necker cube illusion	34
2.4 Hill climbing method	36
2.5 Input line drawings	37
2.5.1 Object projections vs. object drawings	37
2.5.2 Input line drawings vs. 3D object	40
2.6 Implementation of the 3D reconstruction system (SRLD)	41
2.7 Summary	42
Chapter 3 Investigation of Compliance Function in 3D Reconstruction from 2D Sketches	43
3.1 Introduction	43
3.2 Literature review	44
3.2.1 NNs application in 3D representation, recognition and	45

reconstruction	
3.2.2 SVM applications in 3D representation, recognition and reconstruction	47
3.3 Problems in formulating the compliance function	48
3.3.1 Analysis of the regularities in the 3D reconstruction	49
3.3.1.1 Regularity form	49
3.3.1.2 Influence analysis of the regularities in the 3D reconstruction	50
3.3.2 Problem formulation	51
3.4 Introduction of the proposed method	52
3.5 Support vector regression estimation	54
3.5.1 ϵ - Insensitive loss function	55
3.5.2 Support vector regression (SVR)	56
3.5.3 Kernel functions	58
3.6 Generalization performance and parameters tuning	60
3.6.1 Evaluation criteria	60
3.6.2 Parameters tuning	61
3.7 Library set up and regularity preprocessing	62
3.7.1 Library set up	62
3.7.2 Regularity preprocessing – normalization	63
3.8 Experimental results	64
3.8.1 Experimental results from optimization	64
3.8.2 Experimental results from reconstruction	65
3.9 Conclusions	68

Chapter 4 Regularity Selection for Effective 3D Reconstruction	70
4.1 Introduction	70
4.2 Regularities for selection	72
4.3 Literature review	74
4.3.1 Literature review of feature selection	74
4.3.2 Feature selection under SVM/SVR	77
4.4 Regularity selection	78
4.4.1 Bayesian estimation and Gaussian process	79
4.4.1.1 Bayesian estimation	80
4.4.1.2 Gaussian process	81
4.4.2 Automatic relevance determination	81
4.4.3 Estimation of parameters	83
4.4.4 Model selection and validation	85
4.4.4.1 Model selection	85
4.4.4.2 Model validation	85
4.4.5 Support vector regression estimation	86
4.4.6 ARD procedure	86
4.5 Results and discussions	88
4.5.1 Results and discussions from ARD procedure	88
4.5.2 Results and discussions from 3D reconstruction	91
4.6 Conclusions	94
Chapter 5 A Hybrid Intelligent System for 3D Reconstruction	95
5.1 Introduction	95

5.2	An overview of the hybrid intelligent system based on NNs	98
5.3	A hybrid intelligent system for 3D reconstruction	102
5.4	Line drawing categorization.	104
5.5	2D perceptual features extraction and formulation	106
5.6	Classification of line drawings by SVM	109
5.6.1	Binary classification by SVM	110
5.6.2	Multi-classification by SVM	112
5.7	3D reconstruction experts	115
5.8	Procedure of building a hybrid 3D reconstruction system	117
5.9	Experimental results and discussions from 3D reconstruction	117
5.9.1	3D reconstruction results obtained from the different reconstruction experts	117
5.9.2	3D reconstruction error and time comparison	120
5.9	Conclusions	123
	Chapter 6 Topological Analysis of a Single Line Drawing	124
6.1	Introduction	124
6.2	Problem statement	126
6.2.1	Related concepts	127
6.2.2	Principles	129
6.2.3	Topological analysis foundations	130
6.2.4	Discussions of the line drawing representations	130
6.3	Topological analysis of line drawings	131
6.3.1	Dangling edge and dangling face detection and deletion	132
6.3.2	Detecting non-manifold edge and non-manifold vertex	134

6.3.2.1 Non-manifold vertex detection	134
6.3.2.2 Non-manifold edge detection	135
6.3.3 Non-manifold decomposition	137
6.3.4 Possible manifold decomposition and internal face identification	139
6.4 Flowchart and the discussions of the topological analysis	142
6.5 Conclusions	144
Chapter 7 Conclusions and Recommendations	145
7.1 Conclusions	145
7.2 Recommendations for future work	147
References	150
Appendix A Support Vector Regression Estimation	164
A.1 Definition of regression	164
A.2 KKT conditions in SVR problem	165
A.3 Training algorithm	166
Appendix B Object Library	169
B.1 Class 1	169
B.2 Class 2	171
B.3 Class 3	172
B.4 Class 4	173
Appendix C Reconstruction Results from Optimal Regularity Set	174
Appendix D T-test for 3D Reconstruction	176
D.1 T-test for reduced regularity set.	176

D.2 T-test for hybrid intelligent system	177
Appendix E SVM Theory in Classification Task	179
E.1 Basic theory of SVM	179
E.1.1 Linear SVM	179
E.1.2 Nonlinear SVM	181
E.2 Network structure and training method	182
E.3 Generalization performance and evaluation criteria	184
E.4 Selecting training data and tuning parameters	186
E.4.1 Training data selection.	186
E.4.2 Parameters tuning	187
Appendix F Pseudo Code for Classification	188
Appendix G Graph Decomposition	191

List of Figures

Figure 1.1	An idealization of the design process.	4
Figure 2.1	An input sketch is converted to be a proper line drawing.	27
Figure 2.2	A reconstructed 3D object from different viewpoints.	27
Figure 2.3	Information flow and processing scheme in 3D reconstruction system.	28
Figure 2.4	Skewed facial symmetry.	30
Figure 2.5	The process of searching the z -coordinates.	33
Figure 2.6	The Necker cube illusion.	34
Figure 2.7	Ill-reconstruction result caused by the Necker cube illusion.	35
Figure 2.8	Adding artificial lines to indicate coplanarity.	35
Figure 2.9	An isometric projection and an isometric drawing of a cube.	38
Figure 2.10	Various possible input drawings of a rectangular block.	39
Figure 2.11	A 2D sketch window.	41
Figure 2.12	A 3D reconstruction window.	41
Figure 3.1	A pictorial illustration of ε -insensitive loss function.	56
Figure 5.1	(a) An axis-aligned object and (b) a semi-axis-aligned object.	96
Figure 5.2	An input line drawing.	97
Figure 5.3	A typical mixture of experts.	101
Figure 5.4	The architecture of the proposed hybrid system.	103
Figure 5.5	Category of general objects based on the 2D perceptual cues.	104
Figure 5.6	Object examples of different classes.	105
Figure 5.7	Soft Margin Hyperplane.	110
Figure 6.1	The circuit (V_5, V_6, V_7, V_8) is an internal face.	125
Figure 6.2	Incremental reconstruction in the online system (SRLD).	126

Figure 6.3	Line drawings that can be partitioned.	129
Figure 6.4	A pyramid with a dangling edge and face (graph G_1).	132
Figure 6.5	A non-manifold vertex detection.	134
Figure 6.6	A non-manifold object with non-manifold edge.	136
Figure 6.7	Non-manifold edge detection.	136
Figure 6.8	A non-manifold object partition along the non-manifold edge.	137
Figure 6.9	The decomposition of a manifold line drawing.	141
Figure 6.10	Flowchart of the topological analysis of a single line drawing.	143
Figure 7.1	A revised hybrid intelligent system for 3D reconstruction.	149
Figure E.1	SVM Configuration.	182

List of Tables

Table 3.1	Experimental results from optimization process.	65
Table 3.2	Selected experimental reconstruction results applying 2 nd order polynomial kernel and linear kernel to the compliance function.	66
Table 3.3	Normalized standard deviation error for the constructed models.	68
Table 4.1	Commonly used regularities in 3D reconstruction.	73
Table 4.2	Results from ARD.	89
Table 4.3	Reconstruction performance and time saving comparison between the regularity sets S1 and S2.	92
Table 4.4	Selected reconstruction results from different viewpoints.	92
Table 5.1	Optimal regularity sets for different reconstruction experts.	116
Table 5.2	Reconstruction result of objects in Class 1 obtained from the reconstruction expert 1.	117
Table 5.3	Reconstruction result of objects in Class 2 obtained from the reconstruction expert 2.	118
Table 5.4	Reconstruction result of objects in Class 3 obtained from the reconstruction expert 3.	119
Table 5.5	Reconstruction result of objects in Class 4 obtained from the reconstruction expert 4.	119
Table 5.6	Reconstruction performance and time saving comparison between the reconstruction experts in the hybrid system and the benchmark.	121
Table 6.1	Vertex adjacent matrix G_d of Figure 6.4.	133

List of Symbols

ARD	automatic relevance determination
$a_i, a_i^*, \gamma_i, \gamma_i^*$	Lagrange multipliers
b	bias
C	regularization parameter
$\text{Cov}[f(\mathbf{x}_i), f(\mathbf{x}_j)]$	covariance function
CAD	computer aided design
CBIR	content-based image retrieval
Cl_{jk}	binary classifier
CR	connection ratio
CV	cross-validation
d	polynomial kernel function parameter
D	dataset
$d(v)$	degree of a vertex v
E	edge set
$E(f)$	the edges of a face f
E_1	reconstruction error obtained from regularity set S1
E_2	reconstruction error obtained from regularity set S2
e	edge
err_i	normalized reconstruction error
f	face
$F(Z)$	compliance function
$F(\mathbf{x})$	estimator output
$f(\mathbf{x})$	estimation function
$g(\mathbf{x})$	decision function

g_{ij}	the connection information between vertex i and j
G	a graph of a drawing
G_d	vertex adjacency matrix
G_i	sub-graph
GA	genetic algorithm
H	feature space
H_i	model hypothesis
H_{opt}	optimal model
$i(e)$	an interior points of an edge e
$J(\mathbf{f}, \alpha, \gamma, \zeta)$	Lagrangian function
$\mathbf{K}(\mathbf{x}, \mathbf{x}_i)$	kernel function
κ_0	power of $f(\mathbf{x})$
κ_b	bias
κ_l	relevance of the l -th element of $\mathbf{x}$ to the practical output
$L_D(\alpha)$	objective function
$L_z(y, F(x))$	ε -insensitive loss function
l	dimension of regularity vectors
l_0	unit vector of standard line
l_i	unit vector of the line
MAP	maximizing a posterior probability
ME	the mixture of experts
<i>mean</i>	the mean deviation of the regularities
MLP	multiple layer perception
MSDA	minimum standard deviation of angles
NMSE	normalized mean square error

NNs	neural networks
nSV	number of support vectors
p_i	sub-feature extracted from a selected group of entities
p	probability of a regularity
P	statistical index of a 2D feature
$P(D H_i)$	evidence of dataset D under H_i
$P(D \mathbf{f}, H_i)$	the likelihood function
$p(\theta \mathbf{X})$	posterior probability
$P(\mathbf{f} H_i)$	prior probability of $\mathbf{f}$ for a multivariate Gaussian process
$P(\mathbf{f} D, H_i)$	posterior probability of $\mathbf{f}$ for a given dataset D
$P(H_i D)$	posterior probability
$P(H_i)$	prior probability of H_i
R	conditional risk
R1	line parallelism
R2	face planarity
R3	skewed facial symmetry
R4	skewed facial orthogonality
R5	isometry
R6	line verticality
R7	corner orthogonality
R8	line orthogonality
R9	minimum standard deviations of angles
R10	line collinearity
R11	face perpendicularity
R12	planarity of skewed chains

RBF	radial-basis function
$R(e)$	rank of an edge
S	number of classes
S_1, S_2	regularity set
$S_{\text{class } 1}$	the regularity set for Class 1
$S_{\text{class } 2}$	the regularity set for Class 2
$S_{\text{class } 3}$	the regularity set for Class 3
$S_{\text{class } 4}$	the regularity set for Class 4
s	step size
std	the standard deviation of the regularities
SMO	sequential minimum optimization
SVM	support vector machine
SVR	support vector regression
T	training time
T_1	reconstruction time with the regularity set S_1
T_2	reconstruction time with the regularity set S_2
T_{Save}	reduced reconstruction time between T_1 and T_2
V	vertex set
V_{adjacent}	adjacent vertex set of a vertex v
V_{internal}	vertex set of an internal face
v_i	a vertex in a drawing
$v(e)$	endpoints of e
$v(f)$	vertex of a face f
W	global balancing coefficient vector
w	weight vector

w_0	bias
w_{lk}, w_{2k}, w_{kl}	weight coefficients
X, Y, Z	coordinates axes
x_i	regularity term
x	regularity value
$\mathbf{x}$	vector of the n different regularities
$\mathbf{x}_i^l$	the l -th element of input vector $\mathbf{x}_i$
$\{\mathbf{x}_i, y_i\}$	training data sample
$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$	given sample set
$y_i - f(\mathbf{x}_i)$	regression error
y_i	actual outputs
z_i	z coordinate of a vertex i
$z_{i\text{constructed}}$	z coordinate of the vertex in the reconstructed model
$z_{i\text{real}}$	z coordinate of the vertex i in the actual 3D model
Δz	depth range of the real 3D model
Z	coordinate vector of all the vertices of an object
3D	three dimensional
2D	two dimensional
ξ_i, ξ_i^*	nonnegative slack variables
$\phi(\mathbf{x})$	nonlinear transformation function
θ	angle between the lines
ε	tube size
∂f	boundary of a face
$\lambda(\theta, \hat{\theta}(\mathbf{X}))$	cost function

Summary

The objective of this thesis is to develop a sketching-based CAD prototype for 3D shape modeling which can effectively and efficiently convert an input sketch into a 3D geometric model. In this thesis, the work concentrates on general, planar objects reconstruction from a single view line drawing. It addresses some key problems in the optimization-based 3D reconstruction.

Firstly, the basic issues in formulating the compliance function are explored under the framework of support vector machine regression (SVR). Secondly, automatic relevance determination, Bayesian framework and SVR are coupled together to form a regularity selection method. An optimal regularity set is established from the commonly used regularities. Thirdly, a hybrid intelligent 3D reconstruction system is built to handle various line drawings. Line drawings are classified into 4 classes using the technique of support vector machine multi-classification. Effective reconstruction is performed by the “reconstruction experts” which are tuned to deal with each class specifically. Finally, the issues in the topological analysis of a line drawing are investigated, especially in distinguishing manifolds from non-manifolds.

The experimental results confirm that the proposed methods can reliably resolve the 3D reconstruction issues. The contribution of this research is to improve the efficiency and effectiveness of 3D reconstruction with the proposed methods by applying advanced artificial intelligence techniques.

Keywords: support vector machine regression, regularity selection, automatic relevance determination, hybrid intelligent system, support vector machine classification, topological analysis, graph decomposition.

CHAPTER 1

Introduction

If two dimensional (2D) sketches can be converted into three dimensional (3D) models automatically, a natural interface for computer aided design (CAD) systems can be produced. First the roles of sketching in the design process and CAD are discussed, followed by a literature review of 3D reconstruction system from 2D sketches. The objectives and the contributions of the research follow. Finally, the organization of the thesis is introduced.

1.1 The importance of sketching

Perhaps the closest to comprehending human's way of thinking and reasoning is through the use of visual thinking models. The study of visual thinking had its beginning in 1969 (Arnheim). It did not gain a high profile in design research until McKim (1980) demonstrated through experimental studies that visual thinking is vital to all branches of design practice. Sketching is one of the powerful tools for visual thinking. It helps to conceive, record and convey design ideas explicitly and overcome the differences in country, culture and time. Though it is informal, an engineer would understand the form, the proportions and the interrelationships of the elements that make up an object quickly and easily.

Schon (1983) suggested that through drawings, designers construct a "virtual world" where a drawing reveals design ideas and rough geometry. Ullman et al. (1990) noted that "... engineers are notorious for not being able to think without

Chapter 1 Introduction

making “back-of-the-envelope” sketches of rough ideas ... Sometimes these informal sketches serve to communicate a concept to a colleague, but more often they just help the idea take shape”. There is evidence that the physical activity, sketching, helps in generating ideas (Goldschmidt, 1994). Stig (1998) pointed out that starting work with CAD was not the optimal way of working during the initial stages of new product development, re-engineering product development, and production tool development. Rodgers et al. (2000) confirmed that freehand sketching was prevalent in the conceptual phase of design and important in the whole design process. Anne et al. (2001) conducted a postal survey on designers’ working habit and found that almost all designers (96%) use sketches to prepare and support CAD work.

In the design process, engineers proceed gradually from approximations in dimensions and shape towards well-defined geometry. Engineering drawing is used as the interface language in this process to make creative products, save time and money. It has been proved to be a powerful and flexible tool, and constitutes the almost universal engineering language. Sketches are representations that will often allow the designer to “try out” a new idea on paper, quickly and economically. The cognitive process of drawing a line with a pencil is markedly different from that of specifying its endpoints by mouse-clicks. Sketches can provide insight into the designer’s mode of thinking at any particular point in the design process. It is believed that this reflexive analysis will provide a basis to support more effective and efficient use of the designer’s time and effort.

Jenkins and Martin (1992) indicated that rough sketching was important in terms of flexibility and speed and suited the short-term memory of the brain. They emphasized the fact of the reduced cognitive load obtained from sketching by avoiding the need for conversion to formal drawings. Lipson and Shpitalni (1995) summarized the

importance of sketching as fast, flexible, analytical, inexact, abstract, and little commitment. They also concluded that the computers should be made to directly process various sketches produced by the designers.

1.2 Sketching in conceptual design and CAD system

1.2.1 Sketching in conceptual design

The conceptual, preliminary (or embodiment) and detailed design phases together form a process in which a solution for the design problem is generated (Figure 1.1).

The activities performed during these steps are:

- Conceptual design: analyze the requirements, sketch concepts
- Preliminary design: choose materials and dimensions, make renderings
- Detailed design: optimize details, make technical drawings and prototype

Within conceptual design, two distinct yet related activities can be discerned:

- Idea generation: generation and recording of design solutions
- Concept evaluation: simulation and evaluation of design solutions

The early design decisions have a tremendous impact on the final cost of the product. A better choice made earlier can improve the efficiency in the design cycles, reduce unnecessary repeated design phases and avoid costly errors. In practice, a designer has to go back to the previous steps and modify the solutions when new problems occur. Therefore, a designer should improve the quality of the solution, even as early as in the stage of sketching-based conceptual design.

Figure 1.1 An idealization of the design process.

In the conceptual design phase, designers usually generate a number of solutions by sketching them on paper. It is the quickest and easiest way to capture ideas and compare them with each other. Furthermore, looking back at a sketch can inspire different solutions that would not have come into mind if the idea had not been drawn.

1.2.2 Sketching with CAD

According to Ullman et al. (1990), CAD tools aid the designers in four ways: by serving as an advanced drafting tool; through assisting in the visualization of data; by improving data organization and communication; and by pre- and post-processing data for advanced engineering tools such as finite elements or kinematic analysis.

CAD systems are expected to be creative aids for idea generation. Sittas (1991) observed that designers rarely approached a CAD system before they had a well-defined idea of what they intended to do in the conceptual design phase. Designing-by-sketching would be useful considering industrial designers' education, experience and the nature of their work.

Many commercial CAD systems have provided sketching tools, such as AutoCAD, Solid Work, Solid Edge, ProEngineer and Alias Wavefront Studio and so on. The "sketch" in AutoCAD creates hand drawn engineer drawings, and does not support the environment for the conceptual design process. Solid Work and Solid Edge provide a sketching workplace only for saving the temporary drawings with many constraints. The "sketch" in Alias Wavefront Studio is a NURBS based modeler. It is flexible in processing the images and rendering and is powerful for generating and manipulating complex surfaces to realize the designers' imagination. But this tool still remains in the drafting phases, and cannot support conceptual sketching for product design. Moreover, the existence of a sketch is aimed for the detailed design and could not help to capture the designers' inspiration and shape their ideas. In short, "sketch" serves as an intermediate medium between embodiment design and formal drawing.

Taking a closer look at these CAD tools, we would find that the sketching functions advertised are limited to hand drawn drafting, which are more likely to be interactive techniques such as scribbling and painting with different types of tools on

Chapter 1 Introduction

a selected plane, rather than true freehand sketching for design. In all CAD-based-sketching system, the user effectively have to supply all the necessary design data, while the system does not help in deducing the data. In this sense, the ‘D’ in ‘CAD’ really stands for drafting or solid modeling rather than design. It seems that the CAD systems today do not support sketching in any meaningful way yet. The essence of sketching is lost when the CAD users indulge in the drafting tools.

Walderon and Walderon (1988) concluded that “it is the computer that is driving the conceptual designer rather than the other way round”. Lipson (1998) pointed out that the current CAD systems placed a mental load on the designer, and would only distract and impede the designers to capture their inspiration. Clearly, the “sketch” function in the current CAD systems highlights the gap between the CAD system and the creative cognition of designers, rather than being the bridge between visual thinking and realization.

We cannot help asking about the possibility of incorporating the essence of sketching into CAD in the conceptual design phases. This work is termed computer aided sketching for conceptual design. An ideal design support system would be built through integrating the non-digital external representation – sketch – with the digital CAD representations in the early design phase. We will set up the system based on automatic processing and easy communication. That means the designer will focus on visualizing the mental image itself and the system would convert a sketch into a 3D object without being burdened by the interface gadgets.

Nevertheless, for sketches to become the meaningful link between the users and CAD systems, some limitations must be overcome. The sketching-based CAD system is required to accept and understand designs from a sketch or scribbling, which is often vague, inaccurate and sometimes erroneous. The underlining design ideas in a

drawing, from the general information, such as dimensions, manufacturing requirements and material etc., to the specific details such as design or manufacturing features, are needed to be understood, processed and transformed in the sketching-based CAD system. Furthermore, without asking for too much detail, such a system would reason and deduce the design ideas, generate the digital models, evaluate the feasibility of the rough designs and provide advice for possible modifications and improvements.

1.3 Literature review

In order to better understand the literature here, the **recognition**, **interpretation** and **reconstruction** process are defined first. **Object Recognition** means finding the most plausible object from a predefined model library. Gesture-based systems which will be discussed in Section 1.3.2, usually apply this technique to build the 3D geometry of an object. **Feature recognition** refers to finding useful predefined patterns in an object definition or representation. **Interpretation** refers to finding the meaning of a group of entities in a drawing, such as lines, faces or junctions, where lines intersect. Without accurately recovering the 3D scene geometry, the correlations in the entities are extracted to describe what the object is or what it looks like. Therefore interpretation relates to the topological analysis and feature extraction of a line drawing. Identifying real faces from a 2D line drawing and labeling the corners in a drawing are typical interpretation processes. Furthermore, extracting and deducing various geometry properties between the 2D entities and the possible 3D configurations are also categorized as interpretation. **Reconstruction** is the metric recovery of the scenes depicted, which means to derive the geometric information from a line drawing. A reconstruction relies on the prior interpretation due to the

application of the information obtained from this process; but it does not necessarily rely on the prior recognition. Interpreting line drawings properly is an important step in the automatic 3D reconstruction from a single view line drawing.

1.3.1 An overview of sketching-based system

Sketching-based geometric modeling systems provide more natural ways to specify 3D geometric models in contrast to typical WIMP (Windows, Icons, Menus, and Point-and-click) interaction with the current commercial CAD systems. The current sketching-based computer aided design system focuses on 3D shape recovery because a 2D sketch needs to be converted to 3D first. Then, some evaluation and analysis tools, most of which are developed for 3D objects, can be used to produce detailed technical documents. Probably the earliest sketching CAD system is “Sketchpad” developed by Sutherland (1963). When a light pen is used to draw on a screen, graphic primitives such as arcs and lines are created in the CAD system. Since then, sketching systems have demonstrated the potential for applying gestures or line drawings in a geometric modeling system. Moreover, sketching-based CAD systems are turning from the computer centric to human centric. That means products can be designed in a way more natural to the working style of designers. The initial designs are displayed directly in the form of rough wire frame models.

Currently, according to the input pattern, interface and 3D reconstruction methodology, 3D modeling by sketching systems can be classified into gesture-based and line drawing-based sketching systems. The former is based on the calligraphic interface that uses gestures and pen-input to command the modeling operation. The latter is drawing-based geometric reconstruction, developed from the methodology of computer vision, machine intelligence and so on. Moreover, the drawing-based

reconstruction system uses the algorithm to reconstruct geometric objects from a single sketch that depicts its 2D projection. Both systems contain interpretation and reconstruction, but apply different principles and strategies in scene reconstruction. They are reviewed in the following two sub-sections.

1.3.2 Gesture-based sketching system

A gesture-based sketching system uses gesture languages to drive a modeler to perform the reconstruction from a sketch. A gesture is one or more line strokes depicted by a mouse or a tablet stylus on a sketch pad, specifying certain commands. Primitives, such as box, wedge etc., and some particular operations are encoded in various gestures. A gesture-based system works more like a sketching symbol recognition system with predefined gesture library for 3D modeling. A few gesture-based sketching systems are studied to explain their modeling functions and explain why they are not employed in our sketching-based conceptual design system.

Egglı et al. (1995) described a 2D to 3D modeling system named “Quick-sketch”. The details of an object are added gradually to the sketching plane. Moreover, every sketching stroke is always constrained on the plane of the object to avoid a problematic interpretation. A graph-based constraint solver is adopted to establish the geometric relationships and to manipulate the object parts through the interface. Zeleznik et al. (1996) built “SKETCH” system for the architectural shapes. A block-type primitive is defined by three segments starting from the same point. “Sketch-N-Make” allows users without manufacturing expertise to create dimensioned designs and to generate manufacturing process plans from initial designs (Bloomenthal et al., 1998). This system transforms the geometry into a high level feature-based model.

Chapter 1 Introduction

The sketching interface adopts certain mechanisms to constrain and parameterize 3D geometry for precise part design.

Igarashi et al. (1999) proposed a surface modeling system named “Teddy”. This system builds 3D polygonal freeform surfaces from sketched 2D silhouettes. “GIDeS” allows a single-view projection or several orthographic projections as input (Pereira et al., 2000). This system supplies a set of gestures to identify some basic modeling primitives such as a prism and a pyramid. It also defines some modeling operations, such as extrusion and revolution. The dynamic recognition of these modeling gestures enables the proposed system to solve ambiguities. Oliver et al. (2000) described a multi-layered architecture within virtual environments based on sketching-interactions. Specific sketches were adopted to generate cylinder, sphere and torus without ambiguity. Qin et al. (2000) proposed an online sketching system based on fuzzy knowledge. This system can infer users’ sketching intentions and generate the corresponding geometric primitives. Topology information is received dynamically from a 2D sketched input. The introduction of knowledge system is a good start for interpreting a line drawing in the level of features.

A 3D model can be generated through defining and manipulating certain gesture languages. A gesture-based sketching system is good for avoiding ambiguity in the reconstruction process, but it cannot provide a natural interface for the designers to generate rough solids freely. A knowledge-based system is required to contain all possible features and parts for a reasonable interpretation. More rules and knowledge are needed to generate any possible shape conceived by a designer. The gesture language and the hierarchy of system would be complex because various designers in different fields may have specified requirements. Therefore, its application in

practical design is restricted and may not support conceptual design adequately and flexibly.

1.3.3 Line drawing-based interpretation and reconstruction system

Three main techniques are adopted in the line drawing-based reconstruction systems. The first is based on a line labeling scheme (Sugihara, 1982; Grimstead, 1997). The second approach deals with reconstruction as an optimization problem (Marill, 1991; Lipson and Shpitalni, 1996; Company et al., 2004). The third applies a hybrid technique, using the previous two techniques for reconstruction (Varley and Martin, 2000a).

1.3.3.1 Line labeling approach

Line labeling is a basic and well-developed method of assigning meaning to the lines in a drawing. There are two main components in the line labeling-based method: junction catalogue and interpretation algorithm. They are reviewed separately as follows.

- **Junction catalogue**

A junction catalogue is the possible line combinations in a junction and is built to label drawings. The first systematic catalogues were proposed by Huffman (1971) and Clowes (1971) separately. Their catalogues are based on the two assumptions: (1) a line drawing depicts a polyhedron with trihedral vertices; (2) all the hidden lines are removed. They showed that a limited number of junction types are the only possible views of trihedral vertices.

Many other researchers have made contributions to extend the labeling schemes, such as Turner (1974), Sugihara (1978), Chakravarty (1979) and Varley and Martin

(2001) so on. Lee et al. (1985) extended the line labeling scheme to objects with curved surfaces. Malik (1987) proved the validity of a new extended scheme for a general class of objects, such as objects with curved surfaces. Other labeling schemes tried to extend previous schemes to support more general cases and more general objects, such as objects in the Origami world (Kanade, 1980), objects with shadows (Waltz, 1972), objects with dashed lines that represent hidden edges (Sankar, 1977; Sugihara, 1978), non trihedral solids (Huffman, 1977), dashed hidden lines in the Origami world (Marti et al., 1993), curved objects with smooth edges (Cooper, 1999; Cooper, 2000), objects in which lines are added gradually (Grimstead, 1997), and objects with known vanishing points (Parodi et al., 1998) .

In addition, Kanatani (1990) has suggested that non-trihedral vertices can be decomposed into trihedral vertices by splitting them with the labeling. To verify a valid labeling corresponding to a geometrically realizable object, Draper (1980; 1981) proposed the sidedness reasoning method which uses the topology of the sketch rather than the geometry information such as dual-space or gradient-space.

- **Algorithms**

Each junction must be allocated to a label listed in the junction catalogue. The proposed algorithms are generally based on constraint propagation, such as Waltz's propagation algorithm (1972), Draper's sidedness reasoning method (1980; 1981), Kirousis and Papadimitriou's algorithm (1988), and Kanatani's set-intersection method (1990). Finding the proper combinations among the labeled lines meeting at a junction is translated into a discrete constraint satisfaction problem, with the constraint that each line must have the same label (Varley and Martin, 2000b).

- **Systems**

Some researchers focus on applying the line labeling method for practical line drawing reconstruction. Sugihara (1984) used a line labeling scheme to produce probable candidate interpretations. A set of linear equalities and inequalities are formed to define the vertex coordinates and plane equations. The solution that satisfies all the equations builds a valid 3D object. Later, he eliminated some redundant equations to relax the strictness for solution (1986). Grimstead and Martin (1995) adopted an incremental labeler approach to interpret a line drawing. A linear equation system is solved through applying the least squares approximation method.

1.3.3.2 Optimization-based reconstruction method

Automatic 3D reconstruction from a single view line drawing is posed as an optimization problem. An objective function is formulated in such a way that its value is at a minimum when the reconstructed object corresponds to the most plausible object of human perception. It is generally composed using a set of regularities, which are also termed image regularities, merits or perceptual cues in the related literatures (Leclerc and Fischler, 1992; Shpitalni and Lipson, 1996; Varley and Martin, 2000a; Company et al., 2004). Each regularity term measures how well an observable geometrical relationship between a set of entities in 2D is preserved in the corresponding entities in 3D. This method does not impose restrictions on the object as the others do. It is one of the most promising methods in automatic 3D reconstruction. The literature review of this method and its related techniques follow.

- **Reconstruction methods**

Marill (1991) pioneered the use of an optimization method to reconstruct 3D objects from a single line drawing. He suggested that all the angles between all pairs

of lines meeting at a junction should be equal, and termed it minimum standard deviation of angles (MSDA), and used a hill-climbing method to search for the optimum z coordinates of object vertices in the resulting objective function. His approach is based on a simple observation that humans tend to interpret line drawings to form the most “simple” result. Simplicity is expressed by MSDA which is used to inflate a flat sketch into a regularized 3D object. Marill’s approach is interesting since it dispenses with the traditional algorithmic interpreting techniques, and offers a new approach of reconstruction. His method is perhaps more closely oriented towards the nature of the human visual system.

Braid and Wang (1991) refined Marill’s approach and used the conjugate gradient approach to improve the optimization procedure. By introducing a slack parameter in the optimization process, Leclerc and Fischler (1992) refined Marill’s method and introduced planarity into the objective function.

Lipson and Shpitalni (1996) have made some promising contributions to automatic 3D reconstruction from a single view line drawing. They emphasized the contribution of topological analysis in a drawing. A face identification method is proposed to obtain the face sets of the depicted object, thus more image regularities can be used when the face information is available (Shpitalni and Lipson, 1996). They proposed a more comprehensive form for the compliance function for their thirteen regularities. Among the proposed regularities, twelve terms are for planar objects and one for cylindrical objects. They also reported that Brent’s minimization was the most promising optimization technique, although it cannot guarantee to achieve a global minimum. Furthermore, their approach can handle curved object reconstruction, such as cylinders or objects that contain cylindrical surfaces.

Oh and Kim (2003) used perceptual reasoning to overcome the geometric uncertainty of geometry recovery. They adopted a progressive strategy to do the reconstruction in accordance with the sketching order of faces. The perceptual reasoning is abandoned once the projection matrix is determined.

Company et al. (2004) built a 3D reconstruction engine to process sketches generated in a calligraphic interface. A beautifier and an optimization-based 3D reconstruction solver are the two main components. The former converts rough sketches into tidied-up line drawings. The latter handles some typical problems in the optimization-based reconstruction process.

- **Face identification**

An object is bounded by faces. If the faces of an object are known, the complexity of the 3D reconstruction will be reduced significantly because many regularities are extracted from the faces of the depicted object. Much progress has been made in the face identification problem from single line drawings over the past two decades. The techniques are classified into topological and geometric approaches separately (Liu and Tang, 2004a).

(1) Topology-based methods

Topology is the study of the properties of space that remain invariant under geometric transformations. A line drawing is viewed as a graph, where vertices and edges in a drawing correspond to nodes and arcs in a graph respectively. Graph and topology theories are thus adopted to identify faces from a single view 2D line drawing. Based on the connectivity of the edges in a drawing, topological approaches are used to find the circuits representing the faces of the object.

Topology terms and definitions related to the problem of face identification in this section and Chapter 6 are explained as follows. More detailed descriptions can be found in (Mäntylä, 1988; Chartrand and Oellermann, 1993).

Graph: A graph is defined as a finite collection of edges e and vertices v . An edge e is the link between two vertices. An edge cannot pass through the other vertices of a graph. A graph G can be written as $G = (V, E)$ with V and E being its vertex set and edge set respectively.

Note that an edge can also be called as an arc or a link and a vertex can be called as a node in the literatures of graph theory.

Planar graph: A planar graph G is a graph that can be drawn in the plane without any two of its edges intersecting.

Embedding: An embedding of a graph G is a representation of G on a surface so that none of its edges intersect. If an embedding of a graph can be realized on a plane or the surface of a sphere, it is termed as a planar embedding.

Path: A path is an alternating sequence of edges and vertices, where all the edges and vertices are distinct.

Circuit: A circuit, also termed cycle, is a closed trail in a graph where all its vertices except the end vertices are distinct.

k-connected: A graph is connected if a path exists between every pair of vertices. A graph is called k-connected if at least k of its vertices and the edges adjacent to them must be removed to make the remaining graph disconnected.

Tree: A tree is a connected graph without circuits.

Spanning tree: A spanning tree of a connected graph G is a tree of G and contains all the vertices of G .

Manifold: A manifold, or more rigorously 2-manifold, is a solid where every point on its surface has a neighborhood topologically equivalent to an open disk in the 2D Euclidean space.

Internal face: An internal face is a face inside an object and do not participate in generating boundary. Its comprising edges are real edges of an object.

Genus: A genus of a closed surface can be considered as the number of holes that pass through it completely. The genus of a graph G is the minimum genus of a surface on which G can be embedded.

Probably the first pure topological approaches were developed separately in (Hanrahan, 1982; Dutton and Brigham, 1983). Their algorithms proposed that if a drawing can be embedded in a plane with a planar embedding algorithm (Whitney, 1982), there exists uniqueness for the face-circuit relationship between an object drawing and its planar 3-vertex-connected graph. Therefore, the resulting regions, the closed circuits in a graph, represent real faces of objects. This method is applicable only for objects with genus 0 and with 3-connected vertices. Ganter and Uicker (1983) proposed another graph-based approach. A set of fundamental cycles was produced from the spanning tree of the input graph. A cycle reduction procedure was then adopted to identify the true faces from the set. This approach was based on two observations: (a) a cycle, if it represents an actual face of the object, has a minimum number of edges and (b) the sum of the edges of all the cycles forming true faces of the object is minimum. The deficiency in this approach is that it still cannot handle objects with holes. Courter and Brewer (1986) and Hojnicky and White (1988) improved Ganter and Uicker's approach by introducing some better cycle reduction schemes. Still, their approaches failed in dealing with the object of genus >0 . Internal faces and invalid faces are also generated in their approach. Bagali and

Waggenpack's approach (1995) is based on an efficient shortest path algorithm for circuit generation. Their algorithm is fast, conceptually simpler and easy to implement, but is limited to 3-connected drawings of genus 0.

Agarwal and Waggenpack (1992) proposed a distinct decomposition approach to extract face topologies. A divide and conquer strategy was used to remove stars (tetrahedra, N-sided pyramids or multiply connected stars) from a drawing. The real faces of the drawing are obtained by combining triangles that are created from the stars. It provides a new method to handle both manifolds and non-manifolds, but is not efficient.

(2) Geometry-based methods

Geometric approaches find faces by using the geometric structure in a wireframe. The approaches classified in this group may also employ some topological information to do the job.

Markowsky and Wesley (1980) proposed the earliest geometry-based method that can handle objects with planar faces. Their algorithm use algebra, point-set topology, and the normals of planes. However, their algorithm requires the 3D coordinates of vertices to calculate the normal of the planes.

Leclerc and Fischler (1992) developed their algorithm based on a set of psychological and geometric assumptions. For example, parallel lines in a line drawing are parallel in space, a line connecting two vertices on two parallel lines is in the same plane with the two parallel lines, many-sided convex closed contours without internal circuits are likely to be faces, a closed nonintersecting circuit without internal lines corresponds to a planar face. Their approach can only deal with simple planar objects.

Shpitalni and Lipson (1996) extended the work by Leclerc and Fisler (1992) and presented a general approach for face identification. Their method treats a projection as an undirected vertex-edge graph. It assumed that a projection is a perfect line drawing with no missing lines or redundant lines. Their method is built upon an observation on face configuration and the face adjacency theorem. The observation, serving as the criterion for the problem, is that given a line drawing, human beings tend to choose a face configuration in which there are as many edges as possible. The face adjacency theorem states that two adjacent planar faces may coexist in the same object if and only if their common edges are collinear. This theorem can be extended to allow nonplanar and smooth faces. The extended theorem states that two adjacent smooth faces may coexist in the same object if and only if their common edges are smooth. Heuristics rules were formulated to find all the real faces from the potential face space. A heuristics algorithm A^* is used to emulate the psychological reasoning process of a human observer. This method can handle a large range of objects that may be manifolds or non-manifolds.

Liu and Lee (2001) used the same criterion and face adjacency theorem as those in Shpitalni and Lipson's research to formulate the problem. A set that contains minimal potential faces is obtained from a depth-first search algorithm. Two pruning conditions are added as constraints to decrease the number of elements in the set to as small as possible. The maximum weight clique technique is used to find the most plausible faces of an object. Moreover, ambiguous solutions that are obtained in the other interpretation approaches are eliminated in the topological analysis stage. Their approach is robust in analyzing manifold and some non-manifold objects. However, some circuits that do not represent actual faces are also identified. Shpitalni and Lipson's method (1996) also has similar deficiency.

The approach proposed by Liu et al. (2002) focuses on the face identification from line drawings representing manifold objects. They formulated the problem based on a property of manifolds: each edge of a manifold is shared exactly by two faces. The two main steps in the approach are searching for circuits from a line drawing and searching for faces from the circuits. In order to speed up the face identification procedure, a number of properties related to manifold geometry in line drawings are developed to identify most of the circuits that are or are not real faces in a drawing, thus reducing the number of unknown circuits in the second searching.

Liu and Tang (2004b) developed a new approach to tackle the face identification problem by a variable-length genetic algorithm (GA) with geometric constraints and a novel heuristic incorporated in the local searching. Incorporated into the local search algorithm are the maximum ranks, geometric constraints imposed by the face adjacency theorem, and a heuristic called minimal edge face phenomenon. The hybrid GA is designed to find solutions by direct search on a line drawing. Their method tackles the two combinatorial problems involved in (Shpitalni and Lipson, 1996) and (Liu and Lee, 2001) simultaneously. Their method exhibits significantly better performance for objects with faces > 30 , both in computational time and memory requirement.

1.3.3.3 3D reconstruction systems applying the hybrid technique

Some reconstruction systems apply the line labeling method to interpret a line drawing, and then use some perceptual cues in the optimization-based reconstruction method to obtain the 3D object. Thus, these systems would benefit from both techniques, such as Digital Clay (Schweikardt and Mark, 2000), Stilton (Turner et al.,

2000), REFER (Contero et al., 2001), and the system proposed by Varley and Martin (2000a; 2000b; 2000c).

Varley (2002) proposed two labeling approaches to interpret more input line drawings. They are set-intersection and relaxation labeling. Furthermore, Varley and Martin (2002) improved the Grimstead's system by detecting symmetry feature and other related regularities from the sketch. Thus, a complete topology and geometry of the hidden part can be produced. The new system enforces exact constraints when reconstructing the geometry of the boundary representation model (Varley, 2002). Currently, this system is one of the few that can interpret sketches with hidden lines removal and produce a plausible solid model.

This study adopts the optimization-based reconstruction method reviewed in Section 1.3.3.2. The background and the related techniques are explained in Chapter 2.

1.4 Research objectives and contributions

The objective of this project is to develop a sketching-based CAD prototype for 3D shape modeling which can effectively and efficiently convert a single view line drawing into a 3D geometric model. Each input drawing is obtained from parallel projection with all the edges and represents a general planar object. Inaccuracies are inherent in user-produced drawings, where the lines may not be straight and the endpoints may not be exactly connected. A sketch can be converted to be a proper line drawing through beautification and tidying (Lipson, 1998; Liu and Lee, 2000; Company et al., 2004). The drawing thus produced is assumed as the projection of an object and fed to a 3D reconstruction algorithm (Lipson, 1998). In my work, 3D reconstruction is discussed step by step: firstly dealing with exact projections, and

Chapter 1 Introduction

then extending to noisy sketches. This thesis focuses on the exact projection drawings. It integrates the previous investigations and addresses some key problems in the optimization-based 3D reconstruction. The results obtained here ultimately will be extended to inaccurate drawings.

In automatic reconstruction of a 3D object from a single view line drawing, a compliance function is widely used to evaluate the acceptability of a reconstruction. However, how to formulate the compliance function to produce desirable reconstruction results has not been fully investigated. Hence, a systematic method is needed to address the issues in formulating the compliance function.

Various regularities have been used in the reconstruction of a 3D object. These regularities are not all equally informative in the reconstruction process. The issue of regularity selection is discussed, and a method is proposed to identify the most relevant regularities for the task.

A single reconstruction system is usually built to provide a general solution for any possible drawings. The more general the systems are, the less successful they will be for specific reconstructions. A method is proposed to categorize various line drawings into 4 classes, decompose the complex 3D reconstruction problem into some specific sub-problems and select well-targeted reconstruction experts for each specific input.

An input line drawing may contain both non-manifold and manifold elements. A method is needed to establish and convert the non-manifold drawings to manifold drawings, if possible.

Case studies are provided to demonstrate the above proposed methods. The experimental results show that the proposed methods yield significant improvements to realize the effective and efficient 3D reconstruction.

In short, the major contributions of this thesis include:

- (1) A systematic method to investigate the issues in formulating the compliance function for 3D reconstruction.
- (2) An effective method to identify the optimal regularity set for optimization-based reconstruction.
- (3) A hybrid system to reconstruct various 3D objects and a classification method to categorize the input sketch.
- (4) A method to identify the non-manifold elements from a drawing and convert the non-manifold object into manifold object, if it exists.

1.5 Organization of the thesis

This thesis is organized in 7 chapters. Chapter 2 introduces the technical background of optimization-based 3D reconstruction method. The related techniques and their usage are briefly discussed.

Chapter 3 addresses three issues related to the formulation of the compliance function: the best form for the compliance function, the overlapping and conflicting information in the regularities in the formulation, and the computation of the coefficients of the regularity terms in the function. Four different forms of compliance function are separately used in the search for an optimal formulation for the 3D reconstruction under the framework of support vector regression estimation (SVR). The best form that produces the most effective and efficient reconstruction is obtained. A systematic method is provided to calculate the corresponding coefficients of regularities with the use of SVR. Error measurements are made possible by the presence of actual 3D models, whose 2D line projections are taken as the input line drawings for the reconstruction.

Chapter 4 investigates the establishment of the most relevant regularities in automatic 3D reconstruction. Various regularities can be extracted from a single-view line drawing, but not all are deployable: certain regularities may correspond mainly to noise, not information; some may overlap with each other or are not too relevant to the reconstruction. This chapter studies these regularities comprehensively, so as to select the most effective set that can give robust and reliable 3D reconstruction. The selection is made through a method called automatic relevance determination (ARD), which employs the Bayesian framework and SVR. The proposed method is able to identify the worst regularities according to their ARD parameters and eliminate them. The effectiveness of this pruning is evaluated by model validation. The regularity set so obtained is effective for general 3D reconstruction. The experimental results show that the regularity set selected can reduce the reconstruction complexity and produce satisfactory reconstruction performance.

Chapter 5 proposes a new hybrid system to recover 3D shapes from any possible drawings. There are two key components in the proposed system: (1) multi-classifying 2D line drawings (2) developing specific reconstruction experts. A set of 2D features is used as the criteria to establish the categorization of possible drawings. Four classes of drawings are formed by using the technique of support vector machine classification. Then regularity selection would yield an optimal regularity set for each class, and specified reconstruction functions are formulated accordingly. Each input drawing would be assigned into the specific class and handled specifically by certain reconstruction expert. Experimental results showed that the proposed hybrid system can significantly improve the reconstruction accuracy and time consumption than that of a single 3D reconstruction system.

Chapter 1 Introduction

Chapter 6 discusses the issues in the topological analysis of a line drawing. Non-manifold entities are identified first, and then manifold objects are obtained through the operations such as deletion or decomposition, if possible. This chapter provides some basic processing means on the topology of rough drawings by any particular users.

Conclusions are given in the last chapter together with the recommendation for future research.

CHAPTER 2

Optimization-based 3D Reconstruction Method from 2D Sketches

Optimization-based 3D reconstruction methods are widely used to convert various sketches to 3D models (Lipson and Shpitalni, 1996; Liu and Lee, 2000; Company et al., 2004). It has been receiving increasing attention in CAD, computer vision and artificial intelligence.

This chapter discusses the optimization-based 3D reconstruction method in full, as it forms the basis of the work in this thesis. It is organized as follows: Section 2.1 shows the processing flowchart of a general 3D reconstruction system. The adopted regularities for 3D reconstruction are introduced in Section 2.2. Section 2.3 is an overview of the optimization-based 3D reconstruction method. The formulation of the compliance function and the Necker cube illusion are presented in this section. The hill climbing searching method adopted in the optimization process and the initial model are described in Section 2.4. Different types of projections of an object in line drawings and their relationship with the actual 3D object are discussed in Section 2.5. Implementation of the 3D reconstruction system (SRLD) is introduced in Section 2.6. A summary is given in Section 2.7.

2.1 Information processing scheme in 3D reconstruction system

Automatic 3D reconstruction from sketches is basically a line drawing processing

system. A geometrically complete and consistent line drawing is fed to a 3D reconstruction algorithm. Some pre-processing is performed to tidy up rough sketches so as to produce clean line drawings, as illustrated in Figure 2.1. The input sketch is converted to a proper line drawing which contains all the edges and vertices of the depicted object. In this study, all the line drawings are assumed tidied-up. The beautifications of line drawings have been investigated by Jenkins and Marin (1992), Lipson (1998), and Company et al. (2004). An investigation of the beautifier is beyond the scope of this study.

Figure 2.1 An input sketch is converted to be a proper line drawing:
(a) a rough sketch; (b) a clean line drawing.

A line drawing is interpreted by the perceptual cues of the human visual system. Various cues are extracted and formulated as different regularities. These regularities are used as the compliances in the 3D reconstruction method to yield the 3D object.

Figure 2.2 A reconstructed 3D object from different viewpoints.

Figure 2.2 shows the reconstruction result of the sketch given in Figure 2.1.

The information flow and processing scheme in the 3D reconstruction is given in Figure 2.3.

Figure 2.3 Information flow and processing scheme in 3D reconstruction system.

2.2 Regularities for 3D reconstruction

One promising advantage of the optimization-based reconstruction method is its ability to emulate the human perceptual ability in interpreting 2D drawings, through the exploitation of “regularities”, which are the perceptual cues subconsciously picked up in human perception of the scene. Regularities describe relationships between or within a set of geometric entities in a sketch that may be carried over to the 3D object. For example, planar faces in a sketch will remain planar in the 3D object, and parallel lines are likely to remain parallel.

Twelve regularities, including line parallelism and face planarity, have been identified for planar objects (Lipson and Shpitalni, 1996). Each regularity is formulated as a mathematical function whose variables are the z coordinates of the vertices present in the entities contributing to the regularity; this function is designed in such a way that it always returns a positive value which diminishes with greater compliance between the entities in 2D and 3D. The expression of the regularity is the product of two parts. The first part is an observable or presumed geometrical relationship obtained from 2D. The second part reflects the spatial relationship of the current 3D configuration corresponding with the first part.

The regularities serve as the constraints in the search for z to produce an object that corresponds to human perception. The meanings of the regularity terms and their identification methods summarized by Lipson and Shpitalni (1996) are explained as follows, as they form the basis of this study.

(1) **Face planarity**

Face planarity estimates how close the vertices in a face are to lying in a plane. It is a true and important regularity which describes the polyhedral surface of a real object. Face identification from a line drawing is a major problem. The detection methods proposed in (Liu and Lee, 2001; Liu et al., 2002) are adopted in this study.

(2) **Line parallelism**

Parallelism is considered to be one main feature of a drawing (Lamb and Bandopadhyay, 1990; Jenkins and Martin, 1992). A parallel pair of lines in a drawing would very likely reflect real parallelism in space. It is a useful regularity for drawings in parallel projection. The problem is to determine the parallel lines. The bucketing algorithm proposed by Varley and Martin (2000a) seems useful. This method determines the parallel line pairs through finding line groups with the minimum angle deviations.

(3) **Symmetry**

Symmetry is also considered to be a key feature of a drawing (Lamb and Bandopadhyay, 1990; Jenkins and Martin, 1992; Langbein et al., 2001; Piquer et al., 2003). Skewed facial symmetry and mirror symmetry are both used in the optimization method (Lipson and Shpitalni, 1996; Piquer et al., 2003; Company et al., 2004). The former denotes a truly symmetrical face in 3D, while the latter describes the 3D model symmetry. Detecting skewed facial symmetry has received extensive

research in (Langbein et al., 2001; Lipson and Shpitalni, 1996). The simplified detection method proposed by Lipson and Shpitalni (1996) is adopted in this study.

If skewed symmetry exists in a polygonal shape, its axis intersects the contour of the polygon at two points, each either a vertex or the midpoint of an edge. Thus, the number of possible symmetry-axis candidates is n , where n is the number of vertices in the contour. Two conditions are needed for skewed symmetry, as illustrated in Figure 2.4. First, the corresponding two vertices on the face contour must be at equal

Figure 2.4 Skewed facial symmetry.

distances from the symmetry axis; secondly, the lines stretched between the two vertices on the contour must form a consistent angle with the symmetry axis.

(4) **Skewed facial orthogonality**

Skewed facial orthogonality indicates orthogonality in the face contours. In a 2D drawing, the boundary edges in a face alternate between the two main directions of the depicted object consecutively. The statistical behavior of the alternating values produced by multiplying the scalar-product and the cross-product of adjacent 2D lines in the sketch plane is used for detection (Lipson and Shpitalni, 1996). The consistent behavior is recognized as skewed facial orthogonality.

(5) Minimum standard deviation of angles (MSDA)

All angles between all pairs of lines meeting at junctions must be similar (Marill, 1991). This is one of the earliest regularities proposed for optimization-based 3D reconstruction. MSDA is particularly useful to inflate a planar drawing to its 3D object.

(6) Corner orthogonality

Corner orthogonality evaluates a special junction of three lines that is a projection of a 3D orthogonal corner in space. The evaluation criterion is that the combination of three lines and its flipping lines span at least 90 degrees.

(7) Line verticality

A line that is parallel to the y-axis of the drawing plane is likely to be vertical in space. All the points in this line have the same z-coordinates (Figure 2.5). Line verticality is concerned with the position of the object in space than its shape. It is more likely to be a drawing habit by most designers.

(8) Isometry

A human observer assumes that the object in a line drawing have regular dimensions in all directions because a drawing is often, but not always treated as an isometric drawing. However, this regularity cannot be measured directly from a drawing due to three factors: (1) the projection axes are not known yet; (2) various projection directions are possible; (3) the input drawings are not drawn accurately.

(9) Line collinearity

Lines collinear in a drawing are collinear in space.

(10) Line orthogonality

All line pairs in a junction except those that are collinear are perpendicular in 3D.

(11) Face perpendicularity

All adjacent faces should be perpendicular.

(12) **Planarity of skewed chains**

This regularity is not formulated in the compliance function directly. It enforces planarity with symmetry or orthogonality in a chain of entities.

Different researchers propose different regularities for 3D reconstruction (Varley and Martin, 2002; Oh and Kim, 2003; Company et al., 2004). However, the new regularities do not provide further geometric information beyond the 12 regularities discussed above. Therefore, the regularities by Lipson and Shpitalni (1996), the most comprehensive reported in the literature, are used for investigation in this thesis. Company et al. (2004) pointed out that some regularities are direct ones for 3D reconstruction, some are indirect, and some are even false. Therefore, these regularities need to be used selectively.

2.3 Optimization-based 3D reconstruction method

Most existing reconstruction approaches are formulated as optimization problems. They are widely adopted because the optimization-based method can tolerate the inaccuracy in a sketch and produce the desired shape (Lipson and Shpitalni, 1996; Company et al., 2004). So far, the work has been concentrating on objects with planar faces, as is the case in this thesis.

2.3.1 Optimization process for 3D reconstruction

The reconstruction method assumes the use of parallel projection in the sketch, which means that the x , y coordinates of all the vertices in the sketch can remain unchanged from 2D to 3D (Figure 2.5). The 3D reconstruction process is then the finding of the set $Z=\{z_i\}$, $i=1\dots n$, where z_i is the z coordinate of vertex i and n is the number of vertices in the sketch. These z -coordinates form the variables in an

Figure 2.5 The process of searching the z -coordinates.

objective function, and their values are obtained through an iterative process which terminates when the objective function is at its minimum. The objective function, also called compliance function, is formulated as a combination of a set of regularities. The constructed model with the minimum function value in the optimization process is believed to be the model we human would perceive (Lipson and Shpitalni, 1996).

2.3.2 Compliance function for the 3D reconstruction

A properly formulated compliance function is one of the key components in the reconstruction algorithms. Lipson and Shpitalni (1996) proposed a comprehensive form for the compliance function for the regularities discussed in Section 2.2. The compliance function $F(Z)$ is designed such that its value diminishes when the reconstructed 3D model matches better with the shape a person perceives. Therefore, the value of the function can be considered as a measure of the error in the shape recovery process. Lipson and Shpitalni (1996) estimated the reconstruction error with a compliance function as given in Equation 2.1, and reduced this error through iteratively changing the z_i values in Z .

$$F(Z) = W^T \mathbf{x} \quad (2.1)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, the vector of the n different regularities. Each element x_i evaluates the square of the linear deviation from compliance for each regularity, and W is the global balancing coefficient vector, each element of which is the weight for the contribution of the corresponding regularity. Z is the coordinate vector of all the vertices of an object. The elements in Z are the variables in $\mathbf{x}$. $F(Z)$ is hence a linear sum of these weighted regularities, evaluating the error between the reconstructed model and the “ideal” model corresponding to human perception.

With an accurate 2D drawing and using an ideal regularity set, the ideal value of the reconstruction error $F(Z)$ would be zero. However, in practice, the input sketch is not accurate and the regularities cannot be computed perfectly, resulting in some positive value in $F(Z)$.

2.3.3 Necker cube illusion

It has been noticed that a line drawing can be interpreted by the human vision system in two ways, as shown in Figure 2.6. In fact, the human vision system alternates between the two interpretations, both of which are correct. Marill’s work (1991) suggested that the two interpretations would be obtained from different starting points. In the reconstruction process, both of them could possibly be produced. A

Figure 2.6 The Necker cube illusion.

prior prediction of the reconstruction result is needed because the Necker cube illusion sometimes would negatively influence the reconstruction, as illustrated in Figure 2.7(b).

Figure 2.7 Ill-reconstruction result caused by the Necker cube illusion:
(a) input line drawing; (b) ill reconstruction result; (c) desired reconstruction result.

To overcome this problem, human interaction would be preferred, such as enforcing the coplanarity between two faces, as illustrated in Figure 2.8. Adding artificial lines to indicate coplanar faces is firstly advised by Liu et al. (2002) for identifying faces in a 2D line drawing representing a manifold objects, and is further used in the reconstruction process.

Figure 2.8 Adding artificial lines to indicate coplanarity.

2.4 Hill climbing method

The process of searching for the z coordinates of vertices while seeking the best reconstruction is an n -dimensional nonlinear optimization problem (Lipson and Shpitalni, 1996), where n denotes the number of vertices. Solving the multiple-dimensional nonlinear optimization problem is in itself an extensive subject. Many methods can be applied to yield the global optimum which corresponds to the most plausible object perceived by the observers. The investigation of the effective searching algorithm is beyond the scope of this research since the interests are focused on exploring reconstruction issues which affect the 3D reconstruction problem globally, such as the formulation for the compliance function and so on.

In searching for the z coordinates of the n vertices, the hill-climbing method is employed to minimize the compliance function. This method is firstly adopted for 3D reconstruction by Marill (1991) and later by Leclerc and Fischler (1992) and Company et al. (2004). This method is a local optimization algorithm, however, experimental results show that its failure rate is similar to a global optimization algorithm, such as simulated annealing algorithm (Company et al., 2004). Moreover, this method is robust and easy to implement. The minimization results can be compared in each iteration process. Therefore, the hill climbing method is adopted in this study.

Assume a step size s is used in the searching process and the coordinate vector $Z = \{z_1, z_2, \dots, z_n\}$ for all the vertices is set to some initial values. $2n$ possible vectors are produced by sequentially adding and subtracting the step size to an element in the vector. These vectors are: $Z_1 = \{z_1 + s, z_2, \dots, z_n\}$, $Z_n = \{z_1, z_2, \dots, z_n + s\}$, $\dots$, $Z_{n+1} = \{z_1 - s, z_2, \dots, z_n\}$, $Z_{2n} = \{z_1, z_2, \dots, z_n - s\}$. The vector Z_i yielding the minimum reconstruction error is selected as the new Z for the next loop. An incremental step size is then used

for the new searching. This process is repeated until the decrease between the current and last reconstruction error is less than a threshold.

The initial model (a model that contains starting z_i values for the optimization process) is important for the reconstruction process. A good initial estimation of the 3D object would improve the rate of convergence and reconstruction result. Two methods have been developed: the first formulates the estimation into the regularity terms and put them into the compliance function for minimization (Leclerc and Fischler, 1992; Lipson and Shpitalni, 1996); the second builds the initial model directly from perceptual cues (Company et al., 2004). This study uses the first method. In our experiment, we set each value of z_i equal to zero while starting the minimization of compliance function and achieve desirable results in all the test models.

2.5 Input line drawings

2.5.1 Object projections vs. object drawings

An object has width, depth and height. When it is depicted on a two-dimensional medium such as a sheet of paper or the computer screen, the object representation can either be an axonometric drawing or projection, as perspective drawing is not the focus of the current work. The distinguishing feature between a projection and a drawing is the unit of measurement used. In a projection, a scale is constructed and this scale is used to replace measurement. In a drawing, however, the standard unit of measurement is used, e.g. mm, etc. Axonometric projections are further divided into three classes based on the number of major axes which are foreshortened equally (Hill, 1990). The scales constructed for isometric, dimetric and trimetric projections are always smaller than the standard units of measurements from which they are

derived. This means that an axonometric projection is always smaller than an axonometric drawing (Figure 2.9). However, the same design information is available in both. Furthermore, the object dimensions are vague and not important in the conceptual design phase. Therefore, an input line drawing can be treated as a

Figure 2.9 (a) An isometric projection and (b) an isometric drawing of a cube.

projection of an object.

Three types of projection are likely to appear in a practical sketching situation. In trimetric projection, the angles between the axes of the projection are unequal. Three separate scales are needed to generate a trimetric projection of an object. In dimetric projection, the angles between the axes of the projection render two of the three to be equal. Two scales are required to draw the outline of an object in dimetric projection. As in isometric projection, the angles between the axes of the projection are equal, i.e. 120° . The scales are the same as for isometric projections.

A designer may not enforce the angles between the axes of the projection strictly in the sketching process. Furthermore, in most cases the object that a designer try to sketch may not yet exist, so usually he cannot take a look at the object to get a sense of what it looks like. As illustrated in Figure 2.10, a designer would feel uncertain himself when asking about the types of projection and the proportions of the object. In such cases, a drawing can be any types of projection. Thus, a 3D reconstruction

Figure 2.10 Various possible input drawings of a rectangular block.

method is required to be able to process various inaccurate and vague drawings. Fortunately, the regularities, except isometry, are applicable to the three types of projections.

The 3D reconstruction process is termed a reverse process of orthogonal projection (Marill, 1991; Lipson and Shpitalni, 1996). However, in oblique projection, the angles between the projectors and the view plane are no longer orthogonal. Oblique drawings display the exact shape of one face, while displaying an object's 3D appearance simultaneously (Hill, 1990). Since the generation of an oblique drawing does not confer with the reconstruction assumption, oblique projection is not discussed in this study. This does not mean a 3D reconstruction method is unable to process this kind of drawings. Instead, an oblique projection is termed as a special case of axonometric projection (Hill, 1990), and can be reconstructed in the way of axonometric drawing. Take a cube as an example. The front face of a cube which displays the exact shape is handled as a distorted rectangle by axonometric projection. Though the accurate 3D geometry preserved in an oblique drawing is lost, basic geometric properties, such as parallelism and orthogonality, would be carried over to the reconstructed object. Thus,

the essence of designing-by-sketching, creating and visualizing the object corresponding to human perception, is retained.

An oblique drawing can also be reconstructed through measuring its dimensions in a drawing and calculating the corresponding coordinates based on the projection rules. For this case, the optimization-based method may not be needed. Establishing the oblique projection and finding its projection matrix would become the key problems to be solved first. This would also be applicable to axonometric drawings if the projection types of the drawings are established. However, the projection matrix that determines the relationship between 2D and 3D has to be computed precisely, which is not easy for a sketch. With this point in mind, the optimization-based reconstruction method would be better for 3D reconstruction.

2.5.2 Input line drawings vs. 3D object

Company et al. (2004) proposed a “beautifier” which converts a rough sketch into a line drawing in which all the lines are connected properly at vertices and all the parallel lines are parallel. The line drawings produced by the beautifier are equivalent to the line drawings in the reconstruction experiments carried out in this thesis, which are obtained from the parallel projection of the objects from arbitrary viewpoints. The regularities are calculated accordingly from the real 3D models and the projections. The issues in the reconstruction algorithm can be examined in full through comparing an actual 3D model and a reconstructed model from its projection.

2.6 Implementation of the 3D reconstruction system (SRLD)

The investigation of 3D reconstruction is based on a system named SRLD (Liu and Lee, 2000) which can reconstruct the 3D surface of the projection of a 3D wire-frame object drew by a user using a mouse. Figure 2.11 shows a 2D sketch window where

Figure 2.11 A 2D sketch window (Liu and Lee, 2000).

Figure 2.12 A 3D reconstruction window (Liu and Lee, 2000).

the endpoints of an object is specified by mouse-clicks, and Figure 2.12 is the 3D display window showing a 3D object with colored surface.

More functions are added to SRLD to carry the investigation for this thesis, such as creating projections from 3D models, generating regularity database from a number of projections and providing evaluation on the reconstructed objects.

2.7 Summary

This chapter explains the basic ideas and techniques used in the optimization-based reconstruction system. The adopted regularities, searching method and line drawing types are discussed. The next chapter will investigate the formulation issues for the compliance function.

CHAPTER 3

Investigation of Compliance Function in 3D Reconstruction from 2D Sketches

3.1 Introduction

From a sketch, each regularity is identified, evaluated and then weighted by a factor proportional to the perceived importance of the regularity, and its value added to the compliance function. This function is therefore a form of characterization of the sketch, and is at its minimum when the best Z is found (See Section 2.3.1 for the definition of Z). However, the characterization is only rough for a noisy sketch. Thus, the final value of Z that gives rise to the minimum value in the function may not produce the minimum in individual regularities. Furthermore, the regularity detection process is fuzzy: a regularity may not be identified correctly. Therefore, the 3D reconstruction may not attain full compliance in practice.

This chapter proposes a systematic method to establish the best form for the compliance function, investigate the influences of the various regularities and calculate their weighting coefficients for the compliance function. The investigation is performed using support vector regression (SVR), a method now well proven in machine learning and pattern recognition (Haykin, 1999).

This chapter is organized as follows: Section 3.2 is an overview of the 3D object reconstruction, representation and recognition using the technique of neural networks (NNs). Section 3.3 sets out the issues and the problems for 3D reconstruction. Section

3.4 explains the proposed method. Section 3.5 deals with the problems under the SVR framework; the optimal form of the compliance function is investigated and a systematic method is proposed to calculate the global coefficients for the regularities. Sections 3.6 and 3.7 are concerned with specific issues of tuning the SVR parameters and the setting up of object libraries for the investigation. Section 3.8 discusses the results and shows examples from the implementation, followed by the conclusion in the Section 3.9.

3.2 Literature review

Ullman (1996) pointed out that high-level image processing involves extracting information associated with an object that is not directly apparent in the image, such as determining what the object is (recognition), or establishing its exact shape and size (reconstruction). Pal and Pal (1993) predicted that NNs would become widely applied in image processing. In the last two decades, much effort has been put into improving 3D reconstruction by applying NNs technique in diverse fields such as image processing, face recognition, medical imaging, reverse engineering and so on. These applications are related because an algorithm or a processing scheme used in one field is often applicable to other (Frangi et al., 2001).

A brief introduction of the most used NNs is given first, followed by an overview of applying NNs in medical imaging, image processing, face recognition and 3D object reconstruction. The application of support vector machine (SVM), the recently proposed novel technique, in image processing and 3D reconstruction is reviewed in the last part of this section.

3.2.1 NNs application in 3D representation, recognition and reconstruction

Egmont-Petersen et al. (2002) summarized two advantages of using NNs in image processing: (1) applicable to a wide variety of problems; (2) easy to use. The previous research focuses on applying NNs, such as multilayer perceptron (MLP) network, Hopfield network and Radial-basis function (RBF) network etc. in image processing (Frangi et al., 2001; Egmont-Petersen et al., 2002). MLP is an important class of neural networks, and has been applied successfully to solve real world problems. Typically, the MLP network consists of a set of sensor units that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. MLP networks usually are trained under supervision with the back-propagation (BP) algorithm. The BP learning consists of a forward pass and a backward pass. In the forward pass learning, the weights of the network are fixed, and an input is applied to the input layer, and its effect is propagated forward through the network layer by layer until a set of output is produced as the actual response of the network. In the backward pass, the weights of the network are adjusted so as to make the actual response of the network move closer to the desired response in a statistical sense.

Hopfield network consists of a set of neurons and a corresponding set of unit delays, thus forming a multiple-loop feedback system. It is typically used for binary classification problems, and may form some useful models in certain situations. But its practical application is limited due to its complex network structure.

The radial basis function (RBF) network is motivated by the locally tuned response observed in biologic neurons. This network consists of an input layer, a hidden layer and an output layer of neurons. At the input of each neuron, the distance between the neuron center and the input vector is calculated. The output of the neuron is then

formed by applying the basis function to this distance. The RBF network output is formed by a weighted sum of the neuron outputs and the unity bias.

Srinivasan et al. (1993) trained a modified Hopfield network to perform the reconstruction of computerized tomography images. Meyer and Heindl (1998) used MLP regression to reconstruct images from electron holograms. Wang and Wahl (1997) trained a Hopfield network to reconstruct 2D images from pixel data obtained from projections. NNs have been widely used in medical imaging and recognition such as screening for cervical (Cenci et al., 2000) or breast tumors (Brake et al., 2000; Lee et al., 20001). Zhao et al. (2003) gave a comprehensive review of the NNs' application in face recognition. One early example is the probabilistic decision-based neural network method (Lin et al., 1997) and the other is the evolution pursuit method (Liu and Wechsler, 2000). In reverse engineering, the geometry of a real part is scanned to create a CAD/CAM representation with the help of NNs (Hoffmann and Várady, 1998; Yang and Lee, 2000). Face recognition and reconstruction are closely related with reverse engineering because a human face is usually handled as a free form surface. So the two fields are reviewed together as follows.

NNs have been proposed to dynamically fit surfaces to spatial data (Gu and Yuan, 1995). Three 4-layer NNs were designed and trained using a modified BP algorithm. A number of new points are generated by the trained network, providing the computer representations of the existing surfaces. Wei and Hirzinger (1996) employed MLP to reconstruct an object shape from shading. Nezis and Vosniakos (1997) presented a feature recognition system, in which a neural network is used to classify data obtained from a B-rep solid modeler's data base into feature classes, such as pockets, slots, passages, protrusions and so on. Jezekiel and Dibyendu (1998) proposed a multiresolution framework for the reconstruction of smooth surface shapes from

shading images. A BP algorithm is used to learn brightness patterns and associate them with range data.

Riesenhuber and Poggio (1999) used RBF network in their object representation and categorization. They claimed that their scheme is related to recent psychophysical observations on the human perception. Barhak and Fischer (2001) used self organizing map method to create a 3D parametric grid. Their neural network grid converges to the sampled shape through an adaptive learning process and avoids the projection anomalies common to other methods.

Yann and Bahram (2001) presented a two-layer neural network to recognize real 3D objects. Their network was used to process the images obtained from phase-shift digital holograms. The presence and orientation of a 3D object are detected from the network. Cho and Chow (2001) proposed a NNs based hybrid reflectance model to tackle the depth from shading problem. They observed outstanding performance when dealing with images from most real world situations. Piperakis and Kumazawa (2001) proposed MLP networks to represent 3D objects. Using the 3D coordinates of an object as input, the network would indicate whether a point belongs to the object or not after training. Lim and Siti (2004) used MLP network to reconstruct and represent some of 3D objects, which are represented as a set of surface points. The performance is reported promising for all the test examples.

3.2.2 SVM applications in 3D representation, recognition and reconstruction

SVM, a new yet well built method in NNs, has been widely used for image processing and object reconstruction in the last few years (Pontil and Verri, 1998; Egmont-Petersen et al., 2002). Osuna et al. (1997) applied polynomial SVM to perform face detection, and observed that its performance was comparable to other

methods. Pontil and Verri (1998) utilized a linear SVM method to recognize 3D objects from single images and achieved excellent recognition rates in all their experiments such as plain, noise-corrupted and shifted images. Wang et al. (2002) used SVM with a decision directed acyclic graph in their human face recognition scheme. Experiments are performed with 6 images from 50 subjects, with pose and expression variations. Recognition rates exceeding 90% are reported.

Content-based image retrieval (CBIR) is one of the active research topics in image recognition, which is required to find content related images in databases or the internet (Zhang et al., 2001). The SVM technique is widely used in CBIR as a relevance feedback method to capture the user's perception through measuring the similarity between two images (Tao et al., 2004). A SVM classifier is utilized to learn from relevant and irrelevant images marked by users. Experimental results show that the input images which are relevant to the given images can be retrieved efficiently and the retrieval performance has been significantly improved by SVM applications (Hong et al., 2000; Chen et al., 2001; Tong and Chang, 2001; Guo et al., 2002; Peng and Heisterkamp, 2003).

There is no report, however, on applying SVM to automatic 3D reconstruction from single view line drawings. The issues of 3D reconstruction are investigated with this new technique in this chapter and the next two chapters.

3.3 Problems in formulating the compliance function

The regularities and the compliance function for 3D planar object reconstruction by Lipson and Shpitalni (1996), the most comprehensive reported in the literature, are investigated in this section first. Then, the problems in formulating the compliance function are posed.

3.3.1 Analysis of the regularities in the 3D reconstruction

3.3.1.1 Regularity form

Each regularity term measures how well an observable geometrical relationship between a set of entities in 2D is preserved in the corresponding entities in 3D. The final contribution of each regularity term is a combination of all the individual sub-regularities extracted from a drawing. Take face planarity as an example. Every face in the drawing is a sub-regularity contributing to the planarity term. Generally, regularity detection is a fuzzy process. For example, it is not possible to determine with certainty how close to being parallel two lines should be before they start to contribute to the parallelism regularity.

Different authors adopt different sets of regularities and formulate each regularity differently (Lipson and Shpitalni, 1996; Varley and Martin, 2002; Oh and Kim, 2003; Company et al., 2004). Existing methods combine the regularities linearly in the compliance function, but with no adequate explanation on why this form is adopted. Thus, the appropriateness of the form of the compliance function is an open issue. Furthermore, different regularities may contain duplicate and conflicting information, and hence their influence in the compliance function also needs to be investigated.

As the value each regularity returns is a measure of the deviation of the term from compliance, the value is required to be non-negative, which can be ascertained through proper formulation. For example, in the measure for parallelism between two lines, the formula $x = p\theta$ is adopted, where x is the regularity value, p the probability of a regularity existing and θ the angle between the lines. Lipson and Shpitalni (1996) squared each term to ensure a positive value, whereas the linear absolute value is taken in this study, which allows us to investigate the various potential forms for the compliance function in a more flexible way.

3.3.1.2 Influence analysis of the regularities in the 3D reconstruction

Different regularities may overlap or interfere with each other. There exists correlation between various regularities. For example, line collinearity is a special case of parallelism (Company et al., 2004) and line orthogonality is a restricted form of corner orthogonality (Varley and Martin, 2002). In skewed orthogonality, the lines of a face must alternate between two main directions. Thus, some lines may be accounted for in the parallelism regularity. Parallelism may also be found in skewed facial symmetry where lines have consistent angles with the axis of symmetry. Therefore, correlation implies some degrees of redundancy between regularities.

The correlation between regularities may affect the reconstruction positively or negatively. For example, two regularities may have mutually beneficial dependency in 3D reconstruction. Planarity and parallelism have a correlation to some extent since a pair of parallel lines determines a plane. Such a correlation is useful because the two regularities enforce the planarity. However, some regularities, such as MSDA and line orthogonality, may conflict with each other. Take a junction of four non-collinear lines as an example. MSDA regularity requires the angles between all pairs of lines at this junction to be similar, while line orthogonality requires all line pairs at a junction to be perpendicular in 3D. Clearly, the two regularities interfere with each other.

Both overlapping and conflicting information may coexist in the same compliance function. Analysis of the relationships between regularities can yield useful cues in understanding their interactions in the reconstruction process.

Let us consider another example to help understand how regularities affect 3D reconstruction. The line verticality regularity assumes that a line parallel to the y-axis in the drawing plane would be parallel to the y-axis of the 3D coordinate system. This

assumption only holds approximately in practice, however, thereby leading to some deficiencies in the reconstruction. Take the octagonal frustum that has slightly slanted side faces (Table 3.2, Model 7) as an example; the non-parallel lines in the front faces may be detected as inaccurate vertical and parallel lines, given their small deviations from the vertical. Thus, two inaccurate regularities, parallelism and line verticality, play an inappropriately strong role in the reconstruction, resulting in an unsatisfactory 3D model as shown in the last column for Model 7 in Table 3.2.

An accurate regularity should have a bigger influence in the reconstruction. The inaccurate ones may generate noise. There are many noise sources, including imperfect input, imprecise assumptions, and ill sub-regularity recognition. Furthermore, overlap and conflict in the different regularities complicate matters and decrease the reconstruction effectiveness. The current methods do not filter out these noises adequately, and duly include their contributions in the compliance function, leading to unsatisfactory results. Therefore, it is necessary to investigate the optimal formulation for the compliance function based on the regularities, taking into account the noises.

3.3.2 Problem formulation

Problems exist with the use of Equation 2.1 in the reconstruction process.

(1) There is no proof that using a linear formulation, as in Equation 2.1, for the contributions of the regularities to the compliance function is reasonable and optimal for the reconstruction. The influence of overlaps and conflicts in the regularities should be accounted for in the compliance function.

(2) The noises should be tempered, if not filtered out, in the 3D reconstruction. Lipson and Shpitalni (1996) adopted a continuous compliance factor to give tolerance

to small inaccuracies in the input sketch and an inverse trigonometric function to achieve linear deviation from compliance. However, neither procedure can avoid noises, which means the reconstruction error would not be zero even for a perfect input line drawing.

(3) In the compliance function, each regularity term is moderated by a global weight coefficient, an element in the vector W in Equation 2.1. The value of this coefficient is determined through experimental observations previously. Thus, a systematic method is required.

3.4 Introduction to the proposed method

Regression estimation means trying to infer the correct functional dependency from noisy data. Linear function is the general way to approximately describe a model or a system. To obtain better results, increasing the complexity of the function seems to be the solution. But it is not really true due to the effect of “over-fitting” in a system, which means that the complexity of the approximation function is set high. Take spline approximation as an example. Schoenberg (1946a; 1946b) combined l -times differentiable piecewise polynomial functions with a set of smoothness constraints. His approach is proved to be effective with the low dimensional functions. But it is limited to four dimensions in real applications because the number of coefficients increases exponentially with the dimensionality of the problem. Therefore, a suitable method is needed for the function establishment. This method is required to best and effectively describe a system from the given data. The given data is possibly incorrect as it is generated with some additional noise. In most of the cases, little is known about the noise. With these points in mind, the selected method

is required to handle the function estimation, while taking into account various noises simultaneously.

NNs are probably the most popular ways of doing high dimensional regression estimation (Smola, 1996). However, two drawbacks are found. Firstly, the architecture of a neural network has to be determined a priori or modified while training by some heuristic. Thus, it would become a difficult combinatorial problem for MLP. The neural network structure obtained may not be optimal. Secondly, parameters regularization is rather limited. Hence, the capacity of the NNs is not fully exploited.

In contrast with the previous NNs methods, SVM is well built for regression estimation (Smola, 1996). SVM was initially developed by Vapnik (1995) for the classification problem with separable data. Later it was improved to solve the regression problem, called support vector regression (SVR). There is no need to determine the neural network architectures before training. Input data of any dimensionality can be handled in the framework of SVR and a unique solution is found after training. Furthermore, the modeling functions can be chosen among a variety of forms. The capacity of NNs can be controlled effectively through tuning parameters that are involved in the functions or the architectures of NNs.

Thus, SVR is selected to estimate the compliance function for 3D reconstruction. The issues presented in Section 3.3 are investigated under this framework. An explanation of the adopted method is given for better understanding as follows.

The suitability of the form of the compliance function is firstly established. This study compares different forms of compliance function in 3D reconstruction under the framework of SVR. Under the SVR model, each regularity term in the linear form serves as the input. Various forms can be used to formulate the relationship between

the inputs and the reconstruction error. The linear, 2nd and 3rd order polynomial and the Gaussian forms are investigated because they are the typical simple forms for model formulation. As in the SVR model, they are recommended for describing the relationship between inputs and outputs (Haykin, 1999). A linear combination of the regularities is the simplest way to express the compliance function. As the regularities can be unsteady and noisy, it is possible that nonlinear combinations, such as 2nd and 3rd order polynomial form could achieve better performance. The 2nd order polynomial form is similar to the compliance function used by Lipson and Shpitalni (1996), except for the addition of the cross terms among the correlated regularities. The Gaussian form has been considered one of the most effective for a smooth relationship between inputs and outputs. Therefore, the four forms are chosen in this study. Each is embedded into the SVR model separately and their performances are compared so as to establish the optimal formulation of the compliance function. The correlations among different regularities are considered in the investigated forms. The proposed method also calculates the corresponding coefficients of regularities based on the available reconstruction data. All the data are used to compute the global optimal coefficient vector under the framework of SVR.

3.5 Support vector regression estimation

Many techniques can be used in searching for the global balancing coefficient vector W , from simple gradient-based methods to artificial intelligence methods such as genetic algorithm or simulated annealing. However, some of them may be susceptible to being trapped in a local minima, slow convergence in large problems, and high computational cost. Meanwhile, noisy data may mislead the search and increase the difficulties in obtaining the global balancing coefficients with these

traditional approaches. A more effective method, SVR, is adopted to overcome these problems. By applying the structural risk minimization principle (Haykin, 1999), SVR not only can produce the global balancing coefficients within a finite number of iterations, but also provide a systematic method to investigate different forms of the compliance function. Furthermore, as an adaptive method, SVR is especially suitable for dealing with noisy data while saving computational efforts. The detailed application of SVR is presented in this section and more information is given in Appendix A.

3.5.1 ε - Insensitive loss function

In regression estimation, a loss function is required to measure the errors between the estimated values and the target values. Vapnik (2000) suggested using an ε -insensitive loss function as given in Equation 3.1.

$$L_{\varepsilon}(y, F(\mathbf{x})) = \max(0, |y - F(\mathbf{x})| - \varepsilon)$$

$$= \begin{cases} |y - F(\mathbf{x})| - \varepsilon & |y - F(\mathbf{x})| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

where ε is referred to as the tube size (see Figure 3.1), y is the target value, $F(\mathbf{x})$ is the estimator output, and $\mathbf{x}$ is the input vector. As illustrated in Figure 3.1, this loss function only penalizes the errors between the estimated values and the target values as long as they are larger than ε , and ignores errors smaller than ε . Thus, this loss function has the potential to cope with imprecise regularities.

Figure 3.1 A pictorial illustration of ε -insensitive loss function

3.5.2 Support vector regression (SVR)

Consider a set of training data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbf{X} \subseteq \mathbf{R}^n$ denotes the input space of a regression problem, and has a corresponding target value $y_i \in Y \subseteq \mathbf{R}$ for $i=1, \dots, N$. N is the size of training data set. The goal of SVR is to estimate a function $f(\mathbf{x})$ that is not only as close as possible to the target value y_i for every $\mathbf{x}_i$, but also as simple as possible for good generalization. In our approach, selected regularities are used as inputs of this regression model. The target values in the training process are chosen according to the good experimental results of 3D reconstruction, and the SVR estimation function denotes the compliance function. The experiment is carried out in the reconstruction system by Liu and Lee (2000). The target value is obtained from a reconstructed object with minimum reconstruction error.

Nonlinear SVR, a practical method to solve complex regression estimation problems, is utilized in this work. The idea of generating nonlinear SVR is to map the original input space $\mathbf{X}$ into a higher dimensional feature space H by a function $\varphi(\mathbf{x})$ and then construct a linear regressive function $f(\mathbf{x})$ in H . The generic nonlinear SVR estimation function takes the form (Haykin, 1999):

$$f(\mathbf{x}) = \mathbf{w} \cdot \varphi(\mathbf{x}) + b \quad (3.2)$$

$\mathbf{w}$ denotes a set of linear weights connecting the high dimensional feature space to the output space, and b represents bias. The goal is to find $\mathbf{w}$ and b so as to minimize the regression risk. With the introduction of a ε -insensitive loss function, the basic problem of this optimization can be formulated as:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (3.3)$$

subject to

$$y_i - \mathbf{w} \cdot \varphi(\mathbf{x}_i) - b \leq \varepsilon + \xi_i$$

$$\mathbf{w} \cdot \varphi(\mathbf{x}_i) - y_i + b \leq \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0, \forall i$$

The first term $\|\mathbf{w}\|^2$ is related to the complexity of the SVR configuration. Two sets of non-negative slack variables ξ_i and ξ_i^* are introduced to measure the deviation of a data point outside the ε tube shown in Figure 3.1. C is the regularization parameter, which determines the trade off between the smoothness of f and the amount up to which deviations larger than ε are tolerated. A large C assigns a high penalty to the loss function, and produces a complex solution. In contrast, a small C corresponds to a small penalty, thereby produces a simple solution.

Accordingly, the constrained optimization problem, Equation 3.3, is solved with the method of Lagrange multipliers, which is constructed as

$$\begin{aligned} L(\mathbf{w}, \xi^*, \xi, a^*, a, \gamma^*, \gamma) = & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) - \sum_{i=1}^N a_i (\varepsilon + \xi_i - y_i + \mathbf{w} \cdot \varphi(\mathbf{x}_i) + b) \\ & - \sum_{i=1}^N a_i^* (\varepsilon + \xi_i^* - \mathbf{w} \cdot \varphi(\mathbf{x}_i) - b + y_i) - \sum_{i=1}^N (\gamma_i \xi_i + \gamma_i^* \xi_i^*) \end{aligned} \quad (3.4)$$

where a_i , a_i^* , γ_i and γ_i^* are Lagrange multipliers, and $\gamma_i = C - a_i$, $\gamma_i^* = C - a_i^*$.

The last term on the right of Equation 3.4 is included to ensure that the optimality constraints on a_i and a_i^* are variable forms. Data points with $(a_i - a_i^*) \neq 0$ are utilized to formulate this regression function, which are called support vectors and account for only a small part of the training data set. In contrast, data points with $(a_i - a_i^*) = 0$ are located inside the ε -tube, and therefore do not contribute to the regression function. $\mathbf{w}$ is defined as:

$$\mathbf{w} = \sum_{i=1}^N (a_i - a_i^*) \varphi(\mathbf{x}_i) \quad (3.5)$$

Finally the regression function in Equation 3.2 is rewritten as

$$f(\mathbf{x}) = \sum_{i=1}^N (a_i - a_i^*) K(\mathbf{x}, \mathbf{x}_i) + b \quad (3.6)$$

subject to

$$\sum_{i=1}^N (a_i - a_i^*) = 0, \quad a_i, a_i^* \in [0, C] \quad \forall i$$

The kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$ can transform the original input vector into high dimensional feature space explicitly without constructing the feature space.

3.5.3 Kernel functions

When applying SVR to minimize the reconstruction error, the first thing is to select a suitable kernel function to formulate the compliance function. To explore the optimal expression between the regularity terms and the compliance function, four kernel functions are used in this study: linear, 2nd and 3rd order polynomial and Gaussian.

(1) linear kernel $K(\mathbf{x}, \mathbf{x}_i) = \mathbf{x} \cdot \mathbf{x}_i$

Substituting this kernel into Equation 3.6, the regression function is

$$f(\mathbf{x}) = \sum_{i=1}^N (a_i - a_i^*) \mathbf{x} \cdot \mathbf{x}_i + b \quad (3.7)$$

$$(2) \text{ d-order polynomial kernel } K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i + 1)^d \quad (3.8)$$

When using the 2nd and 3rd order polynomial kernel in Equation 3.6, the compliance function $f(\mathbf{x})$ can be expressed as 2nd and 3rd order polynomial of regularities.

For a given input $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, Equation 3.6 can be rewritten as Equation 3.9 with the use of the 2nd order polynomial kernel.

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^N (a_i - a_i^*) (\mathbf{x} \cdot \mathbf{x}_i + 1)^2 + b \\ &= \sum_{k=1}^n w_{2k} x_k^2 + \sum_{k=1}^n w_{1k} x_k + \sum_{k=1}^n \sum_{l=k+1}^n w_{kl} x_k x_l + w_0 \end{aligned} \quad (3.9)$$

where w_{1k} , w_{2k} and w_{kl} are the weight coefficients, w_0 is the bias term, n is the number of regularities and x_k and x_l are the separate regularities.

$$(3) \text{ Gaussian kernel sub } K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{\delta^2} (\mathbf{x} - \mathbf{x}_i)^2\right)$$

In this case, $f(\mathbf{x})$ is formulated as a multivariate Gaussian function of regularity terms.

In the 2nd order polynomial form, the cross terms $\sum_{k=1}^n \sum_{l=k+1}^n w_{kl} x_k x_l$ encapsulate overlapping or conflicting information. The coefficients in the compliance functions are given in Equation 3.9. The proposed compliance function is similar to the form proposed by Lipson and Shpitalni (1996) except for the addition of the last three terms in Equation 3.9. Furthermore, using SVR provides a systematic method to calculate the corresponding coefficients of regularities. The weight coefficients are obtained from the training data and can be used for new line drawings.

3.6 Generalization performance and parameters tuning

3.6.1 Evaluation criteria

Generalization performance measures how well a network performs on unseen data after training. This is analogous to the curve fitting problem with regularization – a good curve fit must represent some underlying trend in the data rather than an arbitrarily close fit to the data points without regard to the solution complexity.

For example, too many free parameters within a network may make the network perform well on the training data, but generalize poorly to similar test data. To avoid these problems, NNs' generalization performance is evaluated in the learning process using a statistical metric called the normalized mean square error (NMSE). NMSE measures the deviation between the target and the predicted values as given in Equation 3.10, and a smaller value suggests a better predictor. In this study, good generalization performance implies good reconstruction performance, thus the NMSE value is used to evaluate the reconstruction performance in the shape recovery process.

$$\text{NMSE} = \frac{1}{\sigma^2 n} \sum_{i=1}^n (f(\mathbf{x}_i) - y_i)^2 \quad (3.10)$$

where $\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (f(\mathbf{x}_i) - \bar{y})^2$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}_i)$.

Besides, the number of support vectors (nSV) is also added to evaluate the performance of the regression because the neural network decision time is proportional to nSV. In SVR, support vectors come from elements of the dataset which are considered important for the task, i.e. a regression surface is constructed according to the support vectors. Under a similar NMSE value, a smaller nSV is preferred. Consider an example, if a regression task can be constructed by two sets of

support vectors, A and B, separately with similar NMSE values, and the size of set A is smaller than that of set B, the trained network based on set A is considered more effective and powerful to describe this regression task than that of set B. Set A also yields a shorter decision time.

3.6.2 Parameters tuning

Choosing optimal parameters is an important step in SVM regression design in order to achieve good generalization performance. The following parameters should be tuned in this work: regularization parameter C , kernel function parameter d (polynomial kernel) or δ^2 (Gaussian kernel), and ε in the loss function. All of them affect the complexity of the regression model, so they are tuned to minimize the NMSE value and related performance. In this study, the tradeoff between the lowest NMSE value and the smallest decision time (the smallest nSV) is the key to the parameter selection.

The parameter ε can affect the number of support vectors used to fit the training data. The bigger its value, the smaller the nSV required, resulting in a simpler model. For the same noise level, ε should be smaller for a larger data set. In general, its selection is based on the estimated noise level of the training data (Hastie et al., 2001; Kwok, 2001).

Several methods can be used to tune the parameters C , d or δ^2 : Xi-Alpha bound (Joachims, 2000), approximate span bound (Vapnik and Chapelle, 2000), and Vapnik-Chervonenkis bound (Burges, 1998). However, none could yield a performance as good as k -fold cross validation (CV), which provides a close correlation with the test error (Duan et al., 2003a). K -fold CV, though crude, is inexpensive to compute. Although it is still not a structured way to select the optimal parameters, it is useful

and efficient in practical applications (Haykin, 1999). Hence, k -fold CV is used in this study to tune these parameters.

In k -fold CV, the training data is randomly split into k mutually exclusive subsets (folds) of approximately equal size. The error is obtained by using $k-1$ subsets for training and then tested on the subset left out. This procedure is repeated k times and each subset is used for testing once, then averaging the test errors over the k trials to give an estimate of the expected generalization error. Note that each dataset needs a sufficient number of test samples so that the performance of the test set, i.e. the testing error, can be taken as an accurate reflection of the generalization performance.

3.7 Library set up and regularity preprocessing

3.7.1 Library set up

In this study, a 3D object library is built to investigate the performance of the compliance function. Over a hundred 3D models with varied geometry are contained in the object library. Thus, thousands of line drawings could be provided for exploiting the relationship between 2D and 3D. The key advantage in creating a 3D object library is the presence of an actual model and its projections, which can be used to quantitatively analyze the issues in the 3D reconstruction, such as the optimal formulation of compliance function and regularity set, and the effective reconstruction strategy, thus giving us a measure on how good a reconstruction algorithm is. The object library developed also forms the basis for the research in Chapters 4 and 5. The full object library is given in Appendix B.

A good object library is critical. There is no well-accepted method for selecting suitable objects for the library; these objects need to exhibit geometry features found commonly. Primitive objects such as box, wedge etc., capture some of these common

features. Hence, different primitives are included in the library and then different models from these primitives are built through different geometrical operations. Thus, the object library contains a large number of objects with different geometric features, such as frustum, pyramid, objects with or without orthogonal corners, and objects of irregular shapes. They also contain various features such as through hole, blind hole, groove, and protrusion. The 3D models are built using a CAD system, and each is then projected from an arbitrary viewpoint to produce a line drawing using parallel projection. The only proviso is that each projection should reveal all the edges and vertices of the object. A line drawing that contains coincident edges and vertices is excluded. Several line drawings may depict the same 3D object.

A library contains both training and testing object sets. The training set is utilized to learn the relationship between the regularities and 3D reconstruction error. Most of them are derived from previous publications (Leclerc and Fischler, 1992; Lipson and Shpitalni, 1996; Varley, 2002; Liu et al., 2002; Oh and Kim, 2003; Company et al., 2004). The test set includes different objects with planar faces, such as frustum, house, tank, and object with hole and so on. A part of the test set is shown in Table 3.2. The set is used to validate the learning process, i.e. the effectiveness of the proposed compliance function. Note that the test drawing is not used for training. Furthermore, regularities extracted from these objects are normalized first, and then used as the inputs for SVR estimation. The normalization process is described next.

3.7.2 Regularity preprocessing - normalization

The normalization of input regularities is the containment of the regularity values within a fixed range. It avoids regularity values with large ranges dominating the calculation and minimizes possible numerical problems. It also improves the

computational efficiency of the NNs. Since the actual minimum and maximum values of different regularities are unknown, bipolar sigmoidal normalization, as given in Equation 3.11 is employed to deal with them:

$$y = (1 - e^{-a}) / (1 + e^{-a}) \quad (3.11)$$

where $a = (x - \text{mean}) / \text{std}$; x refers to the input regularity, mean and std are the mean and standard deviation of the regularities respectively.

With this method, input regularities within a standard deviation of the mean are mapped to the almost linear region of the sigmoidal function. This nonlinear transformation can prevent the most commonly occurring values from being compressed into essentially the same values. Thus, all the regularities can be scaled to the typical range $[-1, +1]$, and used as inputs for SVR estimation in the 3D reconstruction.

3.8 Experimental results

This section presents some experimental 3D reconstruction results from Equation 3.9. The related coefficients are computed from the work presented in Section 3.6 and Section 3.7. The former introduces specific measures for SVR, while the latter is about the data preparation.

3.8.1 Experimental results from optimization

Table 3.1 gives two indices, NMSE and nSV, to show the reconstruction performance of four forms of compliance function. The value of NMSE from a 5-fold CV test indicates the size of the optimization error; and the value of nSV is proportional to the reconstruction time.

Table 3.1 Experimental results from optimization process

Kernel type	linear	2 nd order polynomial	3 rd order polynomial	Gaussian
NMSE	0.0711	0.0558	0.0594	0.0794
nSV	41	28	35	45

NMSE: normalized mean square error from k -fold CV test, $k=5$

nSV: number of support vectors

Table 3.1 shows that the 2nd order polynomial form produces the smallest NMSE and nSV values. The former implies the best reconstruction performance and the latter implies the simplest network structure. Thus, an optimization formulation built with these support vectors would yield the most effective reconstruction at the smallest computational cost.

The other three all yield varying degrees of higher NMSE and nSV values, and thus are less efficient for the purpose. The performance of the 3rd order polynomial form is poorer than the 2nd order polynomial form due to its larger NMSE value. The 3rd order polynomial form is also more complex due to its larger nSV. The linear and Gaussian forms both yield larger nSV and NMSE values, which means that they are not as good for constructing a compliance function based on the training dataset.

The reconstruction performances are investigated using linear and 2nd order polynomial compliance function, as given in Equations 3.7 and 3.9 respectively. The results are discussed in the following sections.

3.8.2 Experimental results from reconstruction

Table 3.2 shows the reconstruction results from drawings of twelve objects. The test set shows good performance with the 2nd order polynomial compliance function (Equation 3.9), as shown in Column 3 of Table 3.2. The prefixed weighting coefficients for the regularities obtained from the support vectors and a bias term are

Chapter 3 Investigation of Compliance Function in 3D Reconstruction from 2D Sketches

embedded into the compliance function. There are two types of weighting coefficients, one for the individual regularities and the other for the interactions between correlated regularities. The standard deviation error in the constructed object is calculated with reference to the actual 3D model. The last column shows the objects reconstructed from using a linear kernel in the compliance function. Some results here are undesirable.

The normalized error for each vertex is given in Equation 3.12:

$$err_i = (z_{i\text{constructed}} - z_{i\text{real}}) / \Delta z \quad (3.12)$$

where $z_{i\text{constructed}}$ is the z coordinate of the vertex in the reconstructed model, $z_{i\text{real}}$ is the z coordinate of the corresponding vertex in the actual 3D model, and Δz refers to the depth range of the real 3D model. The normalized standard deviation errors for the constructed models are given in Table 3.3.

Table 3.2 Selected experimental reconstruction results applying 2nd order polynomial kernel and linear kernel to the compliance function.

Model No.	Input Sketch	3D reconstruction result with the 2 nd order compliance function		3D reconstruction result with the linear compliance function
		View 1	View 2	
1
2

Chapter 3 Investigation of Compliance Function in 3D Reconstruction from 2D Sketches

3
4
5
6
7
8
9
10

Table 3.3 Normalized standard deviation error for the constructed models.

Model No	Error of 2 nd order compliance function	Error of the linear compliance function
1	0.027	0.229
2	0.018	0.274
3	0.024	0.243
4	0.060	0.270
5	0.012	0.284
6	0.043	0.279
7	0.057	0.330
8	0.115	0.505
9	0.134	0.560
10	0.057	0.279
11	0.063	0.262
12	0.028	0.300

3.9 Conclusions

This chapter investigates the relationship between the regularities and their influence in the reconstruction process. Four forms of compliance functions are investigated with the use of SVR estimation. Experimental results show that the 2nd order polynomial form is the best. This function is similar to the one proposed by Lipson and Shpitalni (1996) except for the addition of the terms that express regularity correlations and take account of the overlapping and conflicting information among the regularities. The global optimal coefficients of the regularities

Chapter 3 Investigation of Compliance Function in 3D Reconstruction from 2D Sketches

are obtained using the SVR optimization process, and the resulting compliance function achieved good performance for the 3D reconstruction using the library that has been constructed. The error measurements are made possible by the presence of actual 3D models, whose 2D line projections are taken as the input line drawings for the reconstruction.

CHAPTER 4

Regularity Selection for Effective 3D Reconstruction

4.1 Introduction

The regularities evaluate the geometrical relationships between a drawing and a valid 3D object. Different researchers use different regularities and their views on specific regularities also vary. The early 3D reconstruction methods adopt only a few regularities for 3D reconstruction. Kanade (1981) used line parallelism and skewed symmetry, Lamb and Bandopadhyay (1990) used symmetry cues, Marill (1991) used MSDA, and Leclerc and Fischler (1992) used MSDA and face planarity. Then, Lipson and Shpitalni (1996) improved the reconstruction method with using 12 regularity terms for planar objects and one for cylindrical objects. However, a large set of regularities would increase the complexity of the compliance function and decrease the success rate of the reconstruction method (Company et al., 2004). Varley and Martin (2002) recommended the combination of edge parallelism with either corner orthogonality or junction label pairs. They also reported that the face planarity compliance is beneficial and necessary in the reconstruction process. For polyhedral blocks, origami objects and wire frame models, Company et al. (2004) concluded that face planarity, symmetry and line parallelism are the best regularities. Besides, line collinearity, line orthogonality and skewed facial orthogonality can improve the

Chapter 4 Regularity Selection for Effective 3D Reconstruction

resulting shape. They pointed out that regularities such as line verticality, MSDA and so on can be ignored since they provide overlapping or false information.

The regularity set greatly influences the reconstruction result. The reconstruction success depends largely upon how well the input regularities describe the relationship of the entities in 2D and 3D. Extracted regularities are not always equally informative: some may correspond mainly to noise, not information; some may overlap or conflict with each other in the information they provide or are not too relevant to the 3D reconstruction.

The ways of determining the regularity set are based on experimental observations, which are unable to provide adequate proof of sufficiency and effectiveness on the regularity selection. There is the danger that a chosen regularity set may ignore essential regularities and lose generality in practice. No systematic method has been proposed to establish the optimality of a regularity set and evaluate its effectiveness. Thus, regularity selection is still an open issue in 3D reconstruction. It needs to be investigated systematically so that all the regularities and their collective effects are assessed properly, which this chapter sets out to address.

Chapter 4 is organized as follows: Section 4.2 is an overview of feature selection techniques in NNs, followed by feature selection under SVM. Section 4.3 analyzed the commonly used regularities. A detailed introduction to the automatic relevance determination (ARD) approach under the Bayesian framework in a Gaussian process is described in Section 4.4. Section 4.5 gives the experimental results on regularity selection from the ARD process. It also presents and discusses the 3D reconstruction results, followed by the conclusion in the last section.

4.2 Regularities for selection

One of the key steps in the 3D reconstruction algorithm is the extraction of the regularities from a 2D line drawing. Table 4.1 lists the regularities commonly used in 3D reconstruction. Each of them is assigned a code for ease of display and discussion in this chapter and next. The meaning and the extraction process of each regularity are given in Section 2.2. The regularities in the list are the candidates for regularity selection.

Regularities involve the entities in a drawing, such as edges and faces. Line verticality is for individual lines; line parallelism, face planarity, skewed facial symmetry, skewed facial orthogonality, corner orthogonality, line orthogonality and line collinearity encode the spatial relationship among selected entities; isometry, MSDA and face perpendicularity include all the entities. Some regularities such as skewed facial symmetry and skewed facial orthogonality describe the edge relationships a person would observe. Some regularities may be false in some drawings, while true in some others. Some regularities are active in all 3D reconstructions, while some are not.

Furthermore, for each individual reconstruction, some regularities are likely to correlate with each other. Both overlapping and conflicting regularities may coexist in a regularity set (Section 3.3.1). Thus, some regularities need to be removed while others retained to improve the reconstruction algorithm. Furthermore, each added regularity would need extra computation in its extraction from a drawing and extra complexity in the optimization process, since the regularity vector becomes larger. Hence there ought to be an "optimal set" of regularities, which may vary with different types of input drawings. Means are needed to evaluate the influences,

Chapter 4 Regularity Selection for Effective 3D Reconstruction

suitability and reliability of a regularity set in the 3D reconstruction, an issue this chapter sets out to address.

Table 4.1 Commonly used regularities in 3D reconstruction

Index	Regularity term	Reference
R1	Line parallelism	Lipson and Shpitalni (1996), Varley and Martin (2002), Company et al. (2004).
R2	Face planarity	Leclerc and Fischler (1992), Lipson and Shpitalni (1996), Varley and Martin (2002), Company et al. (2004).
R3	Skewed facial symmetry	Lipson and Shpitalni (1996), Company et al. (2004).
R4	Skewed facial orthogonality	Lipson and Shpitalni (1996), Company et al. (2004).
R5	Isometry	Lipson and Shpitalni (1996), Liu and Lee (2000)
R6	Line verticality	Lipson and Shpitalni (1996).
R7	Corner orthogonality	Lipson and Shpitalni (1996), Varley and Martin (2002), Company et al. (2004).
R8	Line orthogonality	Lipson and Shpitalni (1996), Company et al. (2004).
R9	Minimum standard deviations of angles	Marill (1991), Leclerc and Fischler (1992), Lipson and Shpitalni (1996), Varley and Martin (2002).
R10	Line collinearity	Lipson and Shpitalni (1996), Varley and Martin (2002), Company et al. (2004).
R11*	Face perpendicularity	Lipson and Shpitalni (1996).
R12*	Planarity of skewed chains	Lipson and Shpitalni (1996).

* R11 is not used for selection because of its geometric similarity with R6, R7, R8, R9. It is applied only by Lipson and Shpitalni (1996).

* R12 is not formulated in the compliance function for 3D reconstruction and therefore is not involved in the regularity selection.

4.3 Literature review

4.3.1 Literature review of feature selection

Feature selection is a key component in areas such as pattern recognition, machine learning and image processing (Bishop, 1995; Jain and Zongker, 1997; Burges, 1998). In 3D reconstruction from line drawings, the features are the regularities. Feature assessment is both a complex problem and hot area of research in the last decade because it provides means to evaluate the effectiveness of the extracted features.

Many approaches have been investigated (Siedlecki and Sklansky, 1988). Dash and Liu (1997) provided a comprehensive review of feature selection techniques. Jain and Zongker (1997) constructed a hierarchical taxonomy of feature selection algorithms including NNs, GA, beam search and statistical pattern recognition. These reviewed studies supported the use of feature selection in image processing, pattern recognition and machine learning and reported that the performance of the selected feature set is as good as the full feature set.

It is widely agreed that, in data-limited situations, feature selection should be used to reduce the dimension of a data set, and the classifiers or regressors should operate in this reduced subspace (Scott, 1992; Bishop, 1995). This is particularly so if some of the features are highly correlated with other useful features.

The feature selection methods can be viewed from two aspects: the search methods and the evaluation criteria (Goebel and Yan, 2001). Depending on the search method used, feature selection can be categorized into (1) optimal search such as branch and bound algorithms, (2) heuristic search such as decision tree methods, (3) random search such as genetic algorithms and (4) weight-based search such as NNs. The efficacy of each selection has to be evaluated. The measures include metric, entropy, statistic, accuracy, relevance and so on.

The simplest approach is to select features that potentially have strong influence on the reconstruction. This selection can be done straightforwardly when the problem domain is known. However, it cannot handle highly interacting features.

Conventional methods are based on statistical tools. Among them, the most typical methods are the backward elimination, the forward selection and the stepwise methods (Montgomery and Peck, 1982; Sen and Serivastava, 1990). These statistics-based methods generally consist of three major parts: a selection criterion to determine whether one feature subset is better than another; a systematic search procedure to search all possible combinations of features; and a stopping criterion to indicate when the search may be terminated (Bishop, 1995).

The backward elimination method begins with all the features and removes irrelevant features one by one. The forward selection method adopts a reverse processing sequence compared with the backward elimination method. It begins with an empty feature set and adds one or a few features at each step. The features added are deemed to be the best according to the evaluation criterion used. The stepwise method is a combination of the forward selection and the backward elimination methods. It also starts with zero feature and adds features one by one as in the forward selection method. But after each new feature is added, every selected feature is examined to check whether it should be deleted, just as in the backward elimination method. A detailed description of these statistical-based methods is presented in (Sen and Serivastava, 1990).

Although statistical-based feature selection techniques are widely used to select important features in NNs, they suffer from many limitations (Chenoweth and Obradovic, 1995). Firstly, most of them are computationally expensive. The computation time is quadratic in the number of candidate features. Secondly, it cannot

guarantee an optimal feature subset. Thirdly, most feature selection algorithms are “brittle” due to higher-order interactions between features (Vafaie and Jong, 1993). Besides, with these methods, the feature subsets only changed locally. Once a feature is added at an early step of the search process, it cannot be removed although it encodes the same information as some later selected features. In addition, once a feature is removed, it is unavailable for the remaining features although it can be of critical importance when combined with other features. Finally, a large number of parameters are needed to be set a priori, such as the number of features added or removed, the significance level for selecting features and the final feature size. The quality of their feature selection results vary greatly over different data sets.

There are also various feature selection methods existing in artificial intelligence techniques, such as genetic algorithm (GA), NNs and so on. Some research finds that the salience analysis method performs well in feature selection for the MLP network (Belue and Bauer, 1995; Steppe and Bauer, 1996). Yamany et al. (1997) show that GA can select the optimal subset of features in the MLP network.

To remove irrelevant features, the saliency values of various features are calculated from the problem definition first, then a criterion is set to determine how many features would be removed from the whole feature set, such as adding a noise feature to the original feature set and finding the irrelevant features nearest to the one added (Belue and Bauer, 1995), using statistical hypothesis tests (Steppe and Bauer, 1996) or the largest gap method (Zurada et al., 1997) to detect irrelevant features and using a predetermined threshold to prune the features with a lower saliency (Tay and Cao, 2001). In short, most of the feature selection investigation is restricted to MLP network (Brill et al., 1992; Steppe and Bauer, 1996; Dougherty and Cobbett, 1997; McCalley et al., 1998; Kuncheva and Jain, 1999; Kewley et al., 2000). In this chapter,

a SVM-based feature selection method is proposed, where the threshold pruning is adopted as the criteria to remove the irrelevant regularities.

4.3.2 Feature selection under SVM/SVR

In the past few years, there has been an increasing development in SVM, a supervised optimization method in NNs to predict the output of unseen datasets.

Feature selection is also performed in SVM-based pattern recognition, machine learning and image processing (Barzilay and Brailovsky, 1999). Experimental investigations showed that if the domain knowledge permits feature selection, it should be done for all types of learning algorithms including SVM (Barzilay and Brailovsky, 1999). Added features could improve the predictive power of SVM, while redundant ones could degrade its performance.

Tay and Cao (2001) attempted saliency analysis (SA) bonded with SVM to perform feature selection in financial time series forecasting. SA measures the importance of features by evaluating the sensitivity of the output with respect to weights (weight-based SA) or the feature inputs (derivative-based SA). In SVM, the weights lie in a high dimensional feature space rather than the original input space. Thus, the magnitude of the weights is actually a reflection of the importance of the high dimensional feature inputs instead of the original inputs. Hence, the weight-based SA is not applicable to SVM. Derivative-based SA uses a simple backward search to compute the gradient of network outputs with respect to the inputs. Although this method is easy and simple to implement, it cannot provide a desirable performance (Tay and Cao, 2001).

In the framework of SVM, there are also other methods for feature selection. Bradley and Mangasarian (1998) applied SVM with 1-norm regularized term as an

indirect approach of feature selection. Jebara and Jaakkola (2000) proposed the maximum entropy discrimination method to SVM for feature selection. They selected features in the high dimensional feature space instead of the features from the original input space. However, these methods were limited to the linear kernel or kernels whose higher dimensional feature space can be explicitly expressed. Weston et al. (2001) proposed the gradient descent method to select a feature subset with the smallest leave-one-out error bound. This method has been proved to be more useful for SVM classifiers for the given number of features. However, deciding the size of the feature set is still an open issue.

During the last five years in the field of data mining and artificial intelligence (AI), automatic relevance determination (ARD) proposed by Mackay (1992; 1993) and Neal (1996) has been attracting attention because of its lower computational cost and better performance. Penny and Roberts (1999) found that ARD perform well in feature selection for the MLP network. Though many ARD-based feature selection methods are useful, they are restricted to MLP network (Neal, 1996; Van et al., 2001; Amo et al., 2000; Ragg and Gutjahr, 1997). Recently, this method has been extended to SVM/SVR. The kernel functions which map inputs to outputs in the regression or classification problem are optimized with ARD. In this research, ARD coupled with SVR is employed to perform feature selection. This artificial intelligence-based method can carry out regression estimation and regularity selection simultaneously, while other methods cannot.

4.4 Regularity selection

In existing 3D reconstruction methods, the influence of a regularity is accounted for in a compliance function by the multiplication of its value with a weighting

coefficient. However, this influence may vary due to its interactions with other regularities. For example, a weak regularity may be strengthened by such an interaction. Thus, the effectiveness of a regularity cannot be evaluated in isolation. The formulation of the compliance function uses the presumption that all the regularities have equal efficacy in the reconstruction. However, in practice, some regularities may contribute more to noise. Therefore, it is important to pick the most relevant regularity set as the input to the reconstruction algorithm. Good regularity selection can reduce the reconstruction cost by reducing the number of regularities to be extracted and improve the accuracy.

The following section presents a detailed explanation of ARD under the Bayesian framework. Bayesian estimation, Gaussian process and the principle of ARD are introduced first, followed by parameter estimation of the model by maximizing its posterior probability. Here “**model**” refers to the established mathematical relationship between the input regularities and the output reconstruction error. Finally, various models in the candidate sets are compared using a multiple hypothesis test and the one with the highest evidence is chosen. Also discussed is model validation in terms of the selected regularities.

4.4.1 Bayesian estimation and Gaussian process

To obtain good performance, SVR parameters, such as regularization parameters and parameters of kernel functions etc., should be chosen carefully. These “higher level” parameters are usually referred to as hyperparameters, which are indicated on p. 61. The Bayesian method, which integrates the process of model selection and model averaging, is regarded as a suitable tool to determine the values of these hyperparameters.

4.4.1.1 Bayesian estimation

For a set of N random variables $\mathbf{X} = \{X_1, X_2, \dots, X_N\}$ and a related but unknown random parameter θ , Bayesian estimation provides the framework for obtaining an estimate $\hat{\theta}$ as a function of $\mathbf{X}$ subject to a cost function $\lambda(\theta, \hat{\theta}(\mathbf{X}))$. The conditional risk R is given by

$$\begin{aligned} R &= \int \lambda(\theta, \hat{\theta}(\mathbf{X})) p(\mathbf{X}, \theta) d\mathbf{X} d\theta \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \lambda(\theta, \hat{\theta}(\mathbf{X})) p(\theta|\mathbf{X}) p(\mathbf{X}) d\mathbf{X} d\theta \\ &= \int_{-\infty}^{\infty} p(\mathbf{X}) d\mathbf{X} \varphi(\hat{\theta}) \end{aligned} \quad (4.1)$$

$$\text{where } \varphi(\hat{\theta}) = \int_{-\infty}^{\infty} \lambda(\theta, \hat{\theta}(\mathbf{X})) p(\theta|\mathbf{X}) d\theta$$

When the uniform cost function is defined as

$$\lambda(\theta, \hat{\theta}(\mathbf{X})) = \begin{cases} 0 & |\theta - \hat{\theta}(\mathbf{X})| < \varepsilon \\ 1 & \text{otherwise} \end{cases} \quad (4.2)$$

$$\begin{aligned} \varphi(\hat{\theta}) &= \int_{|\theta - \hat{\theta}(\mathbf{X})| \geq \varepsilon} p(\theta|\mathbf{X}) d\theta \\ &= 1 - \int_{|\theta - \hat{\theta}(\mathbf{X})| < \varepsilon} p(\theta|\mathbf{X}) d\theta \end{aligned} \quad (4.3)$$

Bayesian estimation of θ corresponds to minimizing R ; clearly, this is equivalent to minimizing $\varphi(\hat{\theta})$. As shown in equation (4.3), minimizing $\varphi(\hat{\theta})$ corresponds to maximizing a posterior probability (MAP) $p(\theta|\mathbf{X})$ by

$$\hat{\theta}_{MAP} = \arg \max_{\theta} p(\theta|\mathbf{X}) \quad (4.4)$$

Bayesian inference techniques applied to SVM/SVR offer many advantages, such as:

1. Provide probabilistic prediction that is more desirable than that of deterministic prediction.
2. Select hyperparameters using only training data.
3. Determine the relative importance of each input based on a separate coefficient.

4.4.1.2 Gaussian process

A Gaussian process, which is specified by a mean and covariance function, can be thought of as the generalization of a Gaussian distribution over a vector space. The mean is often taken as zero and the covariance function $\text{Cov}[f(\mathbf{x}_i), f(\mathbf{x}_j)]$ expresses the expected covariance of $f(\mathbf{x})$ between the points $\mathbf{x}_i$ and $\mathbf{x}_j$.

A Gaussian process provides a promising non-parametric Bayesian approach and is better than other models because of its simple probabilistic formulation (Williams and Barber, 1998). It not only can infer hyperparameters in Bayesian framework but also predict probabilistic solution.

4.4.2 Automatic relevance determination

The 3D reconstruction problem is formulated as a regression problem in Section 3.5, which estimates the relationship between the regularity vector and the reconstruction error. The input vector $\mathbf{x}$ consists of all candidate regularities. The function $f(\mathbf{x})$ gives the 3D reconstruction error. A dataset including a pool of input vectors $\mathbf{x}_i$ and outputs y_i are computed from the 3D objects in the library, which is described in Section 3.7.1. The deviation between $f(\mathbf{x}_i)$ and the actual output y_i is the regression error, which is used to evaluate the model performance.

In ranking the regularities, a weight is assigned to each individual regularity. This weight is a measure of and proportional to the degree of relevance of the regularity to

the output. A conventional neural network would fail to set the weights of irrelevant inputs to zero because a limited data set will produce some random relationships between these irrelevant inputs and the outputs (Mackay, 1995). The irrelevant inputs will hurt the model's performance, particularly when the inputs are many and the data few. Therefore, the model needs to embody the concept of relevance, so that it is able to retain the relevant inputs and switch the others off. This can be done using a covariance function, which is a measure of the dependency between two random vectors in a dataset. The function is embedded in the ARD method, and its formula relating the outputs and the inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ is (Chu et al., 2003):

$$\text{Cov}[f(\mathbf{x}_i), f(\mathbf{x}_j)] = \kappa_0 \exp\left[-\frac{1}{2} \sum_{l=1}^d \kappa_l (\mathbf{x}_i^l - \mathbf{x}_j^l)^2\right] + \kappa_b \quad (4.5)$$

where $\mathbf{x}_i^l$ denotes the l -th element of input vector $\mathbf{x}_i$, $l = 1 \dots d$, d is the dimension of the $\mathbf{x}_i$, and the indices i, j correspond to the different vectors. The parameter κ_l determines the relevance of the l -th element of $\mathbf{x}$ to the practical output, κ_0 reflects the power of $f(\mathbf{x})$ and κ_b is the bias.

ARD is adopted to determine the relevance of the regularities to 3D reconstruction. It is assumed that initially weight vectors connected to the same input have common prior covariance, and all the covariance have common prior distribution. In the ARD procedure, different parameters under a model and different models with adjusting weight vectors are explored so as to minimize the regression error.

In this model adaptation process, if κ_l becomes large, it means that the corresponding regularity contains more information and less noise, and vice versa. In other words, a regularity with a larger κ_l is more relevant to the reconstruction. A regularity with a lower κ_l is irrelevant or less relevant and may be ignored. This process can force the weights of weak elements of the regularity vector toward zero,

leaving only the important ones. The ARD parameters κ_0 , κ_b and κ_l , $l = 1 \dots d$, are collected as a model hypothesis H_i . Practical experience indicates that it is useful to assume a Gaussian distribution for the prior probability of model f in a data set D (Mackay, 1995).

Under a model H_i , the prior probability of f for a multivariate Gaussian process can be formulated as

$$P(f | H_i) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left[-\frac{1}{2} f^T \Sigma^{-1} f\right] \quad (4.6)$$

where $f = [f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n)]^T$, n is the number of samples, and the element in Σ is $\Sigma_{ij} = \text{Cov}[f(\mathbf{x}_i^l), f(\mathbf{x}_j^l)]$.

4.4.3 Estimation of parameters

The model can be viewed as a blackbox where some hyperparameters can be adjusted to tune the model performance. Under a model H_i , the method of maximizing a posterior probability (MAP) is used to estimate the adjustable parameters to arrive at the minimum of f . MAP estimation updates the prior knowledge of a model using available data. It enables the estimation of parameters asymptotically approaching the true values of parameters. According to the Bayesian rule, the posterior probability of f for a given dataset D can be written as

$$P(f | D, H_i) = \frac{P(D | f, H_i) P(f | H_i)}{P(D | H_i)} \quad (4.7)$$

where $P(f | H_i)$ represents the prior knowledge regarding the distribution of f .

In MAP, the best model of f can be obtained by maximizing $P(f | D, H_i)$. It corresponds to the best compromise that can be achieved between the data set (represented by the likelihood function $P(D | f, H_i)$) and our prior knowledge $P(f | H_i)$ as shown in Equation 4.7. For a given sample set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in R^d$ is the input

Chapter 4 Regularity Selection for Effective 3D Reconstruction

vector (candidate regularities), y_i is the practical output, $P(D | f, H_i)$ can be expressed by the regression error:

$$P(D | f, H_i) = \prod_{i=1}^n P(y_i - f(\mathbf{x}_i)) \quad (4.8)$$

Similarly, the evidence $P(D | H_i)$ can be stated as

$$\begin{aligned} P(D | H_i) &= \int P(D | f, H_i) P(f | H_i) df \\ &= (2\pi)^{-n/2} |\Sigma|^{-1/2} \int \exp(-S(f)) df \end{aligned} \quad (4.9)$$

$$\text{where } S(f) = \frac{1}{2} f^T \Sigma^{-1} f - \sum_{i=1}^n \ln P(y_i - f(\mathbf{x}_i)) \quad (4.10)$$

Thus,

$$P(f | D, H_i) = \frac{\exp(-S(f))}{\int \exp(-S(f)) df} \quad (4.11)$$

Since $P(f | D, H_i) \propto \exp(-S(f))$, its maximum value can be obtained by minimizing $S(f)$. The minimization of $S(f)$ is formulated as follows:

$$\min \frac{1}{2} f^T \Sigma^{-1} f - \sum_{i=1}^n \ln P(y_i - f(\mathbf{x}_i)) \quad (4.12)$$

This constraint optimization problem in Equation 4.12 can be solved by the method of Lagrange multipliers. With the introduction of the soft ε -insensitive loss function, the corresponding Lagrangian function can be constructed as follows (Chu et al., 2003):

$$\begin{aligned} J(f, \alpha, \gamma, \xi) &= \frac{1}{2} f^T \Sigma^{-1} f - \sum_{i=1}^n \ln P(y_i - f(\mathbf{x}_i)) - \sum_{i=1}^n (\gamma_i \xi_i + \gamma_i^* \xi_i^*) - \sum_{i=1}^n \alpha_i ((1-\beta)\varepsilon + \xi_i - y_i + f(\mathbf{x}_i)) \\ &\quad - \sum_{i=1}^n \alpha_i^* ((1-\beta)\varepsilon + \xi_i^* + y_i - f(\mathbf{x}_i)) \end{aligned} \quad (4.13)$$

where non-negative variables α_i , α_i^* , γ_i and γ_i^* are Lagrange multipliers, with $0 < \beta \leq 1$, and $\varepsilon > 0$. Two sets of non-negative slack variables ξ_i and ξ_i^* are

introduced to measure the deviation of a data point outside the ε tube. The optimization of the objective function $J(f, \alpha, \gamma, \xi)$ can be solved by SVR estimation described in Section 4.5.

4.4.4 Model selection and validation

4.4.4.1 Model selection

The Bayesian evidence framework provides a consistent means to select the most probable model, by comparing the different models quantitatively (Mackay, 1991). A model with the largest value of posterior probability $P(H_i|D)$ is chosen from the candidate data sets. According to the Bayes rule, the posterior probability of a model H_i can be written as

$$P(H_i|D) = \frac{P(D|H_i)P(H_i)}{P(D)} \quad (4.14)$$

where $P(H_i)$ represents the prior probability of H_i and $P(D)$ is a constant for all models. Since there is no preference of one model over another, the prior probability of all the candidate models are equiprobable, i.e. $P(H_j) = P(H_k)$, where H_j and H_k are two candidate models. The model comparison is reduced to comparing the values of $P(D|H_i)$ among the candidates. Using Taylor's expansion to approximate $S(f)$, and introducing this result into Equation 4.9, the evidence $P(D|H_i)$ can be obtained. The optimal model H_{opt} is the model with the largest $P(D|H_i)$. This model reflects the best compromise between the model performance and complexity.

4.4.4.2 Model validation

Model validation follows model selection. It verifies the effectiveness and efficiency of the selected model. A model selection process assumes that adequate models are contained in the candidate set. Even if this assumption is false, the model

selection procedure would still choose a model from the candidate set. Therefore, model validation becomes necessary. The cross validation (CV) method is used to identify the validity of the selected model since it is useful and efficient in practical applications (Haykin, 1999). A selected model is deemed adequate when the regression error of the training data set is similar to that of the testing set. A detailed explanation of CV is given in Section 3.6.2.

4.4.5 Support vector regression estimation

SVR estimation is a universal constructive learning procedure based on statistical theory. The basic idea of SVR estimation is to construct a linear regression in a high dimensional feature space with the minimum risk. Section 3.5 gives the detail of using nonlinear support vector regression with ϵ -insensitive loss function to optimize the compliance function for 3D reconstruction. The 2nd order polynomial is utilized as the kernel function to map the data from the original feature space into a high dimensional space. The minimization problem discussed in Section 4.4.3 is solved by the sequential minimal optimization (SMO) method, which shows an effective and efficient learning performance (Shevade et al., 2000). Appendix A.3 presents the details of the SMO algorithm.

4.4.6 ARD procedure

Before proceeding to the next section, the ARD method for selecting the regularities is summarized as follows:

- (1) Pre-train SVR with the regularity set which includes all the regularity candidates, and obtain the initial values of κ_0 , κ_b and κ_l ($l=1, \dots, d$), where d is the dimension of the regularity set.
- (2) Use κ_0 , κ_b and κ_l as the initial values to train SVR with ARD.

- (3) Under an assumed model H_i , evaluate the posterior probability $P(f|D, H_i)$ and estimate the parameters of this model by MAP.
- (4) Adjust κ_0 , κ_b and κ_l by the gradient-based optimization method and calculate their corresponding evidences in model selection.
- (5) Choose the optimal model with the largest evidence $P(D|H_i)$, and collect the corresponding κ_0 , κ_b and κ_l as the new assumed model.
- (6) Repeat Steps 3-5 until the number of evaluation steps equals a predefined value. Rank all κ_l ($l=1, \dots, d$) in descending order.
- (7) Prune the regularities with the lowest κ_l from the regularity set.
- (8) Perform model validation to verify that the pruning in Step 7 will improve the performance of the neural network in most of cases. Then go to Step 9. If the performance is not improved, recover the pruned regularity in Step 7, record the current regularity set and stop the ARD process.
- (9) Use the retaining regularities to repeat Steps 2-8 until the ARD process stops at Step 8.

Note that the predefined value of evaluation steps in Step 6 should be big enough for SVR to achieve a good generalization performance. A large value may over-fit the model obtained, whereas a small value may result in under-fitting. In addition, the regularity with the lowest κ_l contributes little to the model obtained. Therefore, in step 7, one regularity would be deleted in a trial-and-verify test. To save computational cost, if two regularities have similar κ_l , both can be removed in the same pruning process.

4.5 Results and discussion

4.5.1 Results and discussions from ARD procedure

A data point contains a regularity vector and a reconstruction error. The former is computed from a drawing and its 3D object; the latter is established from the experimental 3D reconstruction results. The dataset were partitioned into two parts randomly: 25% of the data were used in the training set, and 75% in the testing set. Thus, two datasets are formed: the training set is used to learn the relationship between the regularities and 3D reconstruction errors, and the testing set is used to validate the selected regularity set.

Hence, the method adopted can avoid data patterns concentrating on specific types of object and can deal with general 3D reconstruction. This partition method is suggested by neural networks' learning theory. It ensures that both the training and testing data cover the data space comprehensively. The quality of the reconstruction results can be validated adequately. In the training process, all the settings of SVR were kept the same.

Three regularity sets are used in the experiment. They are:

$$\mathbf{S1} = \{R1, R2, R3, R4, R5, R6, R7, R8, R9, R10\}$$

$$\mathbf{S2} = \{R1, R2, R3, R4, R9\}$$

$$\mathbf{S3} = \{R1, R2, R3, R9\}$$

where the regularity codes are defined in Table 4.1. S2 and S3 are chosen as a result of initial experiments on S1. The training and testing results under regularity sets S1, S2 and S3 are given in Table 4.2.

Chapter 4 Regularity Selection for Effective 3D Reconstruction

Table 4.2 Results from ARD

Regularity		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	NMSE	nSV	T(s)
S1	κ_l	2.337	1.858	2.091	2.522	0.331	0.530	1.075	0.343	2.874	0.105	0.022	28	0.247
	Rank	3	5	4	2	9	7	6	8	1	10			
S2	κ_l	2.607	2.203	2.195	2.621					2.965		0.019	25	0.218
	Rank	3	4	5	2					1				
S3	κ_l	2.453	1.474		2.903					2.932		0.119	20	0.211
	Rank	3	4		2					1				

NMSE: normalized mean square error.

nSV: the number of support vectors.

T: training time.

In Table 4.2, three regularity sets are obtained from the ARD procedures. The values of parameter κ_l for individual regularities are given accordingly. The rank of the regularities suggests which regularity can be pruned. The regularity ranked lowest is deleted and then model validation is performed to verify each pruning. Three different regularity sets S1, S2 and S3 are used to show the performance of applying ARD. S1 includes 10 candidate regularities, and S2 is S1 with five regularities, R5, R6, R7, R8 and R10, removed. The set S3 is S2 with R4 removed. Three indices, NMSE, nSV and T(s) are used to establish which regularity set is selected. They are explained first, and how these criteria evaluate individual regularity sets are shown next.

(1) The normalized mean square error (NMSE) is a measure of the regression error. It is the deviation between the target and predicted values as given in Equation 3.10.

The NMSE value is an important index because it indicates the reconstruction performance with the different regularity sets. A smaller value suggests a better

Chapter 4 Regularity Selection for Effective 3D Reconstruction

choice. In this experiment, regularity set S2 produces the most suitable model for the reconstruction.

(2) nSV is the number of support vectors involved in formulating the regression function. It indicates the complexity of the network structure and determines the computing cost of regression estimation. A smaller number is preferred because the reconstruction is achieved faster.

(3) The training time T indicates the learning speed of the model obtained. It is used to compare the learning ability of different regularity sets. A small T indicates a more powerful learning ability.

The regularity set S2 yields the best reconstruction performance considering the above three indices. It is the best compromise between the model performance and the regularity set compactness. The ARD parameters of the removed regularities are very small. This indicates that their relevance is weak to the reconstruction. Some of these removed regularities have large overlap with the regularities retained in S2, and are redundant. The pruning is verified in the testing sets where the regression error (NMSE) decreases slightly from S2 to S1. It means that the pruning benefits the performance of model estimation. Further regularity pruning, R3 deletion, is also tried, leading to the set S3. But this removal is not acceptable because the regression error of S3 shows an obvious increase.

Comparing with the regularity set S1, S2 needs less training time and converges faster for all the data points. This indicates that using S2 can reduce the learning time for a regression task. In addition, the value of NMSE of the reduced set S2 is lower than that of S1. This shows that redundant regularities may lower the performance of the learning algorithm.

Furthermore, coupling the ARD method with the SVR and Bayesian framework can effectively reduce the support vectors in regression. In this case, nSV decreases from 28 (Set S1) to 25 (Set S2). The reduced support vector set would yield a better formulation for the compliance function. Thus, S2 is the effective regularity set, where line parallelism, face planarity, skewed facial symmetry, skewed facial orthogonality and MSDA are selected as the regularities for general 3D reconstruction.

4.5.2 Results and discussions from 3D reconstruction

Based on the conclusion of regularity selection, more sampling data from the library were used to carry out 3D reconstruction. Objects with or without orthogonal corners, and objects of irregular shapes are selected for comparison. Results from 23 line drawings are given in Table 4.3.

The normalized error for each vertex is computed by Equation 3.12. Then, the normalized standard deviation errors E_1 and E_2 for the reconstructed objects, which are obtained from S1 and S2 separately, are given in Table 4.3.

Furthermore, the comparison between the reconstruction time from S1 and S2, T_{Save} , is calculated by Equation 4.15:

$$T_{Save} = (T_1 - T_2) / T_1 \quad (4.15)$$

where T_1 and T_2 are the reconstruction time with the regularity sets S1 and S2 respectively.

The reconstruction time shows a decrease from S1 to S2, with a decrease in dimension in the regularity sets. The reconstruction performances of the two regularity sets are compared under the T-test, which is one of the most popular statistical tests in the machine learning literature (Thomas, 1998). It compares the

Chapter 4 Regularity Selection for Effective 3D Reconstruction

difference between reconstruction errors obtained from the regularity set S1 to S2.

The T-test, as shown in Appendix D.1, shows no accuracy loss with the reduced regularity set S2, while the time saving is marked.

Table 4.3 Reconstruction performance and time saving comparison between the regularity sets S1 and S2.

Case	E_1	E_2	T_{Save}
1	0.0081	0.0058	20%
2	0.0043	0.0044	18%
3	0.0039	0.0054	15%
4	0.0826	0.0783	49%
5	0.1878	0.1969	27%
6	0.0064	0.0070	19%
7	0.0470	0.0509	20%
8	0.0435	0.0425	12%
9	0.0290	0.0299	32%
10	0.0198	0.0192	18%
11	0.1404	0.1402	31%

12	0.0471	0.0509	20%
13	0.0951	0.0908	21%
14	0.1117	0.0891	26%
15	0.0492	0.0521	30%
16	0.0403	0.0308	36%
17	0.0298	0.0403	33%
18	0.0826	0.0870	20%
19	0.0232	0.0237	24%
20	0.0041	0.0047	11%
21	0.1220	0.1249	29%
22	0.0398	0.0481	30%
23	0.0341	0.0413	25%

E_1 : reconstruction error obtained from S1.

E_2 : reconstruction error obtained from S2.

Table 4.4 shows the reconstruction result from different viewpoints on a variety of drawings of objects of various types, whose reconstruction errors are listed in Table 4.3 and Appendix C gives more results.

Table 4.4 Selected reconstruction results from different viewpoints

Model No	Input line drawing	Reconstruction result from different viewpoints		
1

Chapter 4 Regularity Selection for Effective 3D Reconstruction

4.6 Conclusions

To realize effective 3D reconstruction, an intelligent regularity analysis method, which integrates regularity selection, Bayesian framework and SVR estimation is designed and implemented. Various regularities can be extracted from a single-view line drawing, but not all are deployable. Relevant regularities are sensitive in the 3D reconstruction process, while less relevant regularities are less so. The relevant regularities form a set which have the maximum effectiveness and generality in dealing with objects of any shape. ARD coupled with SVR was employed to rank various regularities and prune redundant or less relevant ones so as to optimize the compliance function.

Under the proposed library, the proposed method explores the effective regularities in 3D reconstruction. Line parallelism, face planarity, skewed facial symmetry, skewed facial orthogonality and MSDA are selected as an optimal regularity set for general 3D reconstruction. The method would yield a general compliance function when the object library contains all possible objects.

Experimental results show that the proposed method produces an effective regularity set and a better formulation for the compliance function. The reconstruction time decreases by using the new set and the reconstruction performance is satisfactory. As a systematic method, the proposed method can further be used to produce special formulations for specific types of object reconstruction.

CHAPTER 5

A Hybrid Intelligent System for 3D Reconstruction

In automatic reconstruction of 3D objects from single line drawings, most of the existing systems are single-track, containing one general solution for all drawings. Such a general system may not be able to produce the best solution for a drawing with certain dominant features, such as symmetry, orthogonality and so on. This chapter proposes a hybrid intelligent system in which an input drawing is classified using the technique of SVM classification. The effective reconstruction is performed by the “experts”, purposely developed and tuned to deal with the different classes separately.

5.1 Introduction

Automatic 3D reconstruction from a line drawing is an ill-defined and under-constrained problem with a large solution space. Not all the regularities in the compliance function are equally applicable to every input drawing. A regularity's contribution to a reconstruction varies with the characteristics of the object. For example, there is no parallelism in a tetrahedron; facial symmetry is minimal in asymmetrical objects. The regularities inapplicable in a drawing may adversely influence its reconstruction process. But what is inapplicable in one drawing might be applicable in another.

Lipson and Shpitalni (1996) claimed that an observer may identify some trends in the line directions in a sketch. The trends correspond to the main dimensions of the solid body, and thereby suggest the basic shape of an object, such as orthogonal, spherical and so on. Grimstead and Martin (1995) proposed a bucketing algorithm to group lines based on the directions and identify parallel lines. Later, Varley and Martin (2000a) developed a bundling algorithm with improved performance. Furthermore, they categorize the sketches with hidden topology into eight special cases, such as axis-aligned and semi-axis-aligned objects, frustum etc. An object is axis-aligned when every edge is parallel to one of the three coordinate axes (Figure 5.1). They emphasized that axis-aligned and semi-axis-aligned objects are common in engineering practice, but they also pointed out that the two types may not be guaranteed to be common features in the sketches by any particular user. Their

Figure 5.1 (a) An axis-aligned object and (b) a semi-axis-aligned object (Varley, 2002).

taxonomy is useful in deducing the hidden topology and for geometric finishing. But it is sophisticated (Company et al., 2004) and fails to interpret drawings with non-trihedral vertices (Varley and Martin, 2000a). Company et al. (2004) classified polyhedra into normalons, pyramidal and prismatic objects. The normalon, characterized as rectangular trihedral polyhedron, plays a fundamental role in their 3D reconstruction engine. A sketch is marked normalon if the bundling process

successfully identifies exactly three groups of parallel lines. Simple prismatic and pyramidal objects are identified by comparing relative edge-vertex connection information.

It can therefore be seen that different researchers classify line drawings differently. Their drawing classification methods may not account for all possible line drawings due to the diversity and complexity of drawings of objects. The blend of different types of drawing may not be processed with the existing classification methods. Take the drawing in Figure 5.2 as an example, four groups of parallel lines are found, moreover, there are four vertices with four incident edges. This drawing may not be classified into the proposed categorizations. Hence, the classification of various drawings is still an open problem. There also lacks a systematic framework to handle any classified reconstructions comprehensively.

In this study, a hybrid intelligent system that integrates various artificial intelligence techniques, such as NNs and feature selection, is built to handle possible reconstructions. Based on the divide-and-conquer strategy, the task domain of 3D reconstruction is decomposed into several sub-regions. Each sub-region reflects 3D reconstructions from a specific drawing class. The reconstruction configuration best

Figure 5.2 An input line drawing.

fitting in with its class is developed to tackle a task. The “nature” of various possible line drawings is detected first: (1) selecting a set of 2D perceptual rules and formulating them; (2) establishing a categorization based on the rules. Then, within each specific class, regularity selection is adopted to discard the regularities that are irrelevant or less relevant to the specific reconstruction, and the optimal compliance function is formed to capture the relationship between the classified 2D drawings and their 3D models. An optimal sub-solution is obtained for each sub-region, with decreased reconstruction complexity and improved accuracy. Thus, the proposed hybrid system would take advantage of the strengths from different reconstruction methods. An extensive experimental study shows that this hybrid system is effective in improving the reconstruction performance and decreasing the computational complexity.

Chapter 5 is organized as follows: Section 5.2 is an overview of hybrid intelligent system. The incorporation of the SVM into the hybrid system is introduced. The framework of hybrid system for 3D reconstruction is proposed in Section 5.3. Line drawing categorization is introduced in Section 5.4. Section 5.5 presents the 2D perceptual features, which are used to classify the line drawings. A detailed classification of line drawings by SVM is described in Section 5.6. The 3D reconstruction experts for the four classes of drawings are given in Section 5.7. Section 5.8 presents the experimental results, followed by the discussions from 3D reconstruction. The conclusion is given in the last section.

5.2 An overview of the hybrid intelligent system based on NNs

A hybrid intelligent system is a computational system which is based on the integration of different artificial intelligence techniques (also called soft-computing

techniques), such as NNs, data mining, expert systems etc. The strengths of these different techniques are exploited by the hybrid system. Thus, not only the overall system performance is increased, but also the drawbacks of these techniques are eliminated. The hybrid system includes some “expert” networks, which have been trained to solve specific tasks. In recent years, it has growing applications in language understanding (Wermter, 1997), industrial fault diagnosis (Adgar et al., 1998), face recognition (Mirhosseini et al., 1998), tissue identification from magnetic resonance imaging (Chao et al., 1999), CAD system for electrical machine design (Zakarian and Kaiser, 1999) and so on. The hybrid intelligent system has been gaining increasing application due to its adaptive ability in handling vague and imprecise information (Bergan and Trubatch, 2002).

There are many possible combinations among the different soft-computing techniques and knowledge representation methods, and also different ways to integrate them. For example, NNs can be combined with fuzzy logic, case-based reasoning, or GA to produce a unified model or even a co-processing scheme. Much attention has been paid to incorporating the powerful neural network techniques into a hybrid intelligent system in the last decade. The previous research is focused on exploiting the potential of artificial NNs (Bonissone, 2000; Trentin and Gori, 2001); little is done with the recently developed SVM.

A proper structure is needed to encapsulate SVM into the hybrid system. The mixture of experts (ME) structure is introduced first, followed by a discussion on how to generalize SVM into this structure.

The divide and conquer strategy is often used to attack a complex problem by dividing it into some simple sub-problems and seeking for solutions separately. The ME structure is one of the powerful techniques inspired by this strategy and has led to

many simple, elegant and efficient solutions in practice (Jacobs et al., 1991; Jordan, 1994; Kwok, 1998). The motivation of ME is that individual expert networks can focus on specific regions and attack them well. ME consists of a set of expert networks and a gating network that cooperate with each other to solve a problem (Figure 5.3). Specifically, the expert networks are used to handle different input regions which are softly decomposed from the entire input space by a gating network. Then the outputs of the expert networks are combined by the gating network to obtain the solution of the problem.

Based on the idea of using different experts for different input regions, Milidiú et al. (1999) generalized the ME architecture into a two-stage architecture to handle the non-stationarity in the data. In the first stage of their system, a clustering algorithm is used to partition the entire input space into several disjointed sub-regions. Then, in the second stage, a mixture of experts competes to handle partitioned sub-regions. Only the expert best fitting in with each sub-region is used for the final prediction. By taking this strategy, the proposed method has an adaptive architecture in the sense that any model can be chosen as the expert candidate. Furthermore, by applying the most adequate model to each partitioned sub-region, this generalized ME architecture significantly improves prediction performance in comparison with using a single expert model to deal with the entire input space.

Figure 5.3 A typical mixture of experts.

The idea of generalizing support vector machine into the ME architecture was first theoretically discussed by Kwok (1998). Based on the ME architecture in Figure 5.3, Kwok directly used multiple SVMs as expert networks and produced a powerful support vector mixture which is applicable to both classification and regression problems. Tay and Cao (2001) utilized the self-organization feature map to partition an entire input space into several disjointed sub-regions, and had different experts tackle different sub-regions. This method is an unsupervised clustering algorithm based on the competitive learning algorithm (Kohonen, 1989), in which the training data points with similar characteristics in the input space are classified into the same sub-region. Their neural network architecture is similar to the one proposed by Milidiú et al. (1999). For each sub-region, only the SVM expert that is the most adequate is used for the final solution. Based on the architectures discussed here, a SVM-based hybrid intelligent system for 3D reconstruction is proposed in the next section.

5.3 A hybrid intelligent system for 3D reconstruction

Two problems exist in formulating the reconstruction algorithm: the unlimited geometric and topological variations in a line drawing and the limited prior knowledge of the object to be reconstructed. Imperfect drawings and imprecise assumptions on the object further exacerbate the problem. No single general strategy can reconstruct every possible object accurately and effectively. With this point in mind, it might be useful to pre-process line drawings so that we might be able to utilize a more appropriate reconstruction technique, if it exists. The pre-processing refers to the analysis of the geometry in a drawing to identify the dominant geometry properties inherent in it.

Finding a proper solution requires leverage on two kinds of resources: problem domain knowledge and available reconstruction data, such as input drawings and output 3D models. The domain knowledge is obtained from empirical knowledge and the data from reconstruction experiments. Both are incomplete, noisy and sometimes erroneous, thus difficult to apply. The ME architecture, a well developed structure for hybrid systems, is adopted to incorporate these useful data. The underlying idea is that the relationships between a number of input drawings and their 3D models have a more uniform distribution under the partitioned sub-region than the entire object space. An expert would capture the relationship in the reduced domain more easily and effectively. The generalized ME architecture therefore would be more efficient in comparison with a single reconstruction system.

Figure 5.4 shows the architecture of the proposed hybrid system. It includes extracting 2D features from the input drawings, classifying the drawings into separate classes in terms of 2D features, and using the reconstruction experts to obtain the 3D

objects. A 2D feature reflects the statistical information about certain geometric properties existing in the 3D object to be reconstructed. For example, all the symmetrical faces can be identified from a 2D line drawing, thus the feature of symmetry is defined numerically by the percentage of symmetrical faces existing in all its faces. Hence, a drawing would be described explicitly by this feature. Various features can be extracted from a 2D line drawing; each of them describes the object to be reconstructed partially. They work together to interpret an object and are used as the criteria to categorize line drawings. With the usage of the SVM technique, multi-classification is adopted to classify and label the drawings into the specific classes. Furthermore, the multi-classifier of drawings serves as the gating network to tune the cooperation of reconstruction experts.

Figure 5.4 The architecture of the proposed hybrid system.

Despite many different possible inputs, the hybrid system would tackle the reconstructions within several local regions, and the 3D reconstruction experts would always apply the most appropriate regularities and reconstruction formulations for each particular class. The following sections will address the 2D feature extraction, multi-classification of input drawings and the specified reconstruction experts separately.

5.4 Line drawing categorization

Effective 3D reconstruction could be achieved within a specific object domain (Lipson and Shpitalni, 1996). Thus, the classification of 2D drawings becomes necessary. The proposed classification (Figure 5.5) is established based on the geometry of the depicted object that can be deduced from 2D line drawings. Four classes are developed. The first two classes are based on orthogonality, one for objects perceived to possess, and the other not possess, prominent facial-orthogonality. Objects with orthogonality prominent are common in mechanical design (Samuel et al., 1976). Under orthogonality, a further classification is made based on symmetry, for two reasons. Firstly, symmetry is fundamental in human visual perception (Palmer, 1999). Secondly, studies have shown that a large proportion of industrial components have some symmetry (Langbein et al., 2001). Classes 1 and 2 are thus established as illustrated in Figure 5.5. When the faces of an object contain both orthogonality and symmetry, the object would be, to a large extent, axis-aligned. Semi-axis aligned objects would also fall into the two classes. For objects that are not

Figure 5.5 Category of general objects based on the 2D perceptual cues.

prominently orthogonal, parallelism is used as a further classification criterion and Classes 3 and 4 are obtained. Classes 3 and 4 contain objects that have no or few orthogonal faces and corners. The objects in Class 3 have prominent parallelism, while Class 4 is likely to be pyramid-like or irregular in shape. Hence there are four classes of drawings, with some variations admitted within each class.

Figure 5.6 gives examples of the four different object classes, more examples are given in Appendix B. A geometrical property is said to be prominent when it can be detected in most of the investigated entities, such as faces, lines, and corners. Take an object which is bounded by n faces as an example, if m faces are found symmetrical,

Figure 5.6 Object examples of different classes.

and $m/n > 65\%$, this object would be considered symmetry prominent. Similar checks can be made to establish the statistics of the existence of other properties in an object. The class of an input drawing is hence determined by its 3D shape and then used as the training pattern for the SVM classification.

5.5 2D perceptual feature extraction and formulation

Feature extraction is a key component in pattern recognition, image processing and machine learning. It provides valuable information for object representation and recognition (Peters, 1991; Henderson et al., 1994). Though a drawing contains only 2D information, some geometrical properties can be perceived by observers that can possibly carry over to the real 3D object. These observable 2D rules are called 2D features, which are obtained from selected entities such as a pair of lines, a corner or a face. They do not normally appear in a 2D drawing accidentally (though rare, there can be accidental parallel lines, especially when you give a large tolerance), and can be extracted as the characterization of a real 3D object. Feature extraction enables us to establish the taxonomy of various line drawings from their perceived 3D shapes.

There is a lack in adequate computational models to emulate the human perceptual ability. Extracted 2D features often, but not always, reflect true spatial information. However, in practical applications, such as image processing, object recognition etc., various perceptual cues can be organized together to do the recognition and interpretation. These features would form a mechanism to achieve desirable results. An extracted feature is required to correlate with certain possible characteristics existing in an object to be reconstructed (meaningfulness), and its computing task cannot be too expensive (simplicity). In this study, probabilistic assumption is

introduced to define a 2D feature. It describes how well an object can be characterized by certain features statistically.

Take line parallelism as an example. The bucketing method proposed by Varley and Martin (2000a) is adopted to group lines based on their directions. Three line groups would contain the majority of lines. In each group, the line that has the most popular direction is selected as the standard to evaluate the parallelism within this group. The formulation of the feature applies the weight expression proposed by Lipson and Shpitalni (1996). The function $\mu(x)$ is used to evaluate parallelism taking into account the sketching inaccuracy: $\mu_{a,b}(x) = \max(0, 1.1 \cdot e^{-\frac{(x-a)^2}{b}} - 0.1)$, where x represents the value to be checked, a is an ideal value for parallelism ($a=0^\circ$) and b is a selected bias to tolerate the inaccuracy of the parallelism ($b=10^\circ$). Then, $p_{parallel} = \mu_{0^\circ, 10^\circ}(\cos^{-1}(l_i \cdot l_0))$ can compute the possibility of parallelism between two lines, where $p_{parallel}$ is the measure on how close the lines are being parallel, l_0 is the unit vector of the standard line and l_i is the unit vector of the line to be checked.

A set of similar local cues can be extracted from 2D drawings, such as line verticality, corner orthogonality, skewed facial orthogonality, skewed facial symmetry, line collinearity etc. (Lipson and Shpitalni, 1996). Each feature term is a combination of all the individual sub-features extracted from a drawing. Take skewed facial symmetry as an example. This feature evaluates the skewed symmetry existing in a polygonal shape. Every face in a drawing would form a sub-feature contributing to the final term. The final term of a feature is given as Equation 5.1, which reflects the situation of certain features existing in the whole drawing.

$$P = \frac{1}{n} \sum_{i=1}^n p_i \quad (5.1)$$

where P is the statistical index of certain features inherent in the object to be reconstructed, p_i is the sub-feature extracted from a selected group of entities, and n is the number of extracted sub-features.

The above features form the criteria to categorize input line drawings. Some features like skewed facial orthogonality and symmetry are objective information both geometrically and topologically. They form the main criteria in the establishment of the proposed drawing categorizations, for two reasons. Firstly, the faces of a 3D object in a drawing can be identified correctly purely from the vertex-edge connections (Liu et al., 2002). Secondly, the correct extraction of these two face-related features is feasible because the methods of detecting them from a polyhedral shape have been studied extensively (Friedberg, 1986; Posch, 1992; Yip et al., 1994; Shen et al., 2001) and used successfully (Lipson and Shpitalni, 1996; Varley and Martin, 2000b; Piquer et al., 2003).

Moreover, some other features are used in determining the drawing class such as parallelism, corner orthogonality, line collinearity and so on. However, some of them tend to be assumptions and the detections are uncertain or ambiguous. Take corner orthogonality as an example: it cannot be established with absolute certainty, which may adversely influence the classification result. But this does not mean it does not qualify as a criterion for the classification. Corner orthogonality is coupled together with facial orthogonality to evaluate the orthogonality of an object. Our mechanism of deducing the geometric properties of a 3D object from the features extracted from a single view drawing may not correspond exactly with the real mechanism of human perception on spatial objects, but it is useful in our exploitation.

The extracted 2D features serve as the input to train the neural network for SVM classification. Drawings with similar cues are put into the same category. In this way,

an interpretation-by-features procedure is formed to roughly categorize an input drawing from its spatial object.

5.6 Classification of line drawings by SVM

NNs are used for the classification; they include unsupervised and supervised networks. For an unsupervised classification, such as adaptive resonance theory (ART) based method, the training set only consists of input training patterns. The network learns to adapt based on the experiences collected through the previous training patterns. It has weaknesses, such as local minima, over-fitting and so on. The classifier is severely affected by noise points. It may fit the noise and not just the data because no noise removal mechanism is employed (Baraldi and Blonda, 1999). Thus, it imposes a great challenge for feature extraction techniques since extracting sensitive features for the classifier is still an ongoing issue in this research.

Supervised learning networks are the mainstream in the development of neural models. The training data consist of both input and output training patterns. Thus, the learning will benefit from the assistance of the teacher. The network is taught to learn the relationship between inputs and outputs, just like a teacher teaches a student to learn what is correct. The learning is terminated when the network can give correct output for any input. In this study, SVM classification, a supervised method, is applied to learn the class of the drawings using the extracted 2D features as input.

SVM can take advantage of prior knowledge of 3D objects and construct a hyperplane (a linear function that is capable of separating data without error) as the decision surface so that the margin of the separation between different class examples is maximized. Compared with other clustering algorithms, SVM possesses a firm mathematical foundation and excellent performances, such as minimizing the system

complexity and yielding satisfactory classification accuracy. A detailed introduction about SVM classification is given in Appendix E.

In the proposed classification system, a training data set is used to train NNs to perform 2D line drawing classification. The training data set consists of a number of training samples. Every pair of input vector and output value constitutes a training sample. The input vector comprises the features extracted from a 2D line drawing. The output value is the class of this particular drawing. The testing set which has the same constitution as the training set is used to validate the learning process, i.e. the effectiveness of the multi-classification.

5.6.1 Binary classification by SVM

The training samples contain many different drawings. It is impractical to separate the input drawing space by a hyperplane without error, but it would be desirable to do the separation with a minimal number of errors. Figure 5.7 gives a brief illustration of the binary classification algorithm by SVM, where the middle line is the decision surface between the two classes and the outer lines are the limits to the tolerance of the classification error. The decision surface is defined as $g(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}_i) - b$, where

Figure 5.7 Soft Margin Hyperplane.

$\mathbf{w}$ is the weight vector, b is the bias and $\varphi(\mathbf{x})$ denotes a set of nonlinear transformations from the input space to the high-dimensional feature space (Haykin, 1999). The space between the two outer lines ($g(\mathbf{x}) = \pm 1$) is called soft margin. When a sample is located in its class space, e.g. a sample from class 1 located below $g(\mathbf{x}) = -1$ or a sample from class 2 located above $g(\mathbf{x}) = +1$, it has $\xi_i = 0$; otherwise, $\xi_i > 0$, where ξ_i is a positive slack variable representing the deviation of a sample from its outer line. As illustrated in Figure 5.7, x_1 belongs to class 2, but it jumps to the space of class 1 according to the decision boundary $g(\mathbf{x}) = 0$ and has a ξ_1 . Similarly, x_2 jumps to the space of class 2 and has a ξ_2 . They are named non-separable samples and need to be penalized in the classification algorithm.

Hence, the primal problem of the SVM classification algorithm is constructed as follows (Haykin, 1999):

$$\min_{\xi, \mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad (5.2)$$

subject to the constraints $y_i(\mathbf{w} \cdot \varphi(\mathbf{x}_i) - b) \geq 1 - \xi_i$, and $\xi_i \geq 0, i = 1, \dots, n$.

In Equation 5.2, positive slack variables ξ_i are introduced in the constraints to quantify the nonseparable data (located in soft margin) in defining the hyperplane, since the data from 3D reconstruction cannot be strictly separated in a high-dimension feature space. Regularization parameter C determines the tradeoff between the training error and model complexity. A larger C corresponds to assigning a higher penalty to training errors, over-fits the training set and leads to a deterioration in the generalization performance. In contrast, a small C makes the weight placed on the training data so slight that a large classification rate occurs.

Using the method of Lagrange multipliers, the optimization problem in Equation 5.2 is solved by maximizing the objective function $L_D(\alpha)$.

$$L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (5.3)$$

subject to these constraints:

$$\sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function, and α_i is the Lagrange multiplier. The Gaussian kernel, a kernel function commonly used in classification, is adopted to map the feature data from the original space into the high dimensional space (Hsu and Lin, 2002). Its mathematical expression is given in Equation E.7 (Appendix E.1.2.). The decision surface that separates input feature data into two classes is defined by Equation 5.4:

$$\sum_{i=1}^n \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) = 0 \quad (5.4)$$

5.6.2 Multi-classification by SVM

Since the input drawings are to be classified into four classes, a multi-classification is needed. The common multi-classification approaches include (1) all-together, (2) one-versus-all and (3) one-versus-one. The all-together approach attempts to consider the examples from all classes together at once (Vapnik, 2000). However, its training speed was found to be slow (Duan et al., 2003b). The one-versus-all approach constructs S binary classifiers where S is the number of classes. The i th binary classifier is trained with all the examples from the i th class with positive labels, and the examples from all other classes with negative labels. The output of a sample $\mathbf{x}$ is assigned to the class with the largest decision function value. However, when considering the testing accuracy and training time, this approach cannot compete with the one-versus-one approach, which is more suitable for practical usage (Hsu and Lin,

2002). A brief introduction is explained as follows. Appendix F presents the pseudo code.

The one-versus-one approach is implemented with a “Max-Wins” voting strategy. It constructs one binary classifier for every pair of distinct classes. Hence, it constructs $S(S-1)/2$ binary classifiers. The binary classifier Cl_{jk} is trained with examples from the j th class with positive labels and the k th class with negative labels. For example, if classifier Cl_{jk} says $\mathbf{x}$ is in class j , then the vote for class j is increased by one. Otherwise, the vote for class k is increased by one. After every of the $S(S-1)/2$ binary classifiers casts its vote, the output is assigned to the class with the largest number of votes. Although such an approach requires $S(S-1)/2$ classifiers to be trained, the problem set is smaller (as it only involves two classes for each classifier) and usually results in faster convergence during training. Hence, this approach is adopted for the multi-classification of line drawings in this work.

To illustrate the “Max-Wins” voting strategy, let us consider an example of three-class classification, where there are three binary classifiers, Cl_{12} , Cl_{13} and Cl_{23} . The classifier Cl_{ij} reflects the comparison between class i and class j . If $Cl_{12} = +1$, it indicates that the outcome belong to class 1 rather than class 2 and vice versa for the case of $Cl_{12} = -1$. To find the winning class, we can compute the score of each class: $\text{score_1} = Cl_{12} + Cl_{13}$, $\text{score_2} = Cl_{21} + Cl_{23}$ and $\text{score_3} = Cl_{31} + Cl_{32}$ (noting that $Cl_{ij} = -Cl_{ji}$). The classification result is made based on the score values. If the value of score_1 is the biggest, $\mathbf{x}$ is predicted to be class 1.

In the one-versus-one approach, for the training data from the j th and k th classes, the following binary classification problem has been solved:

$$\min \frac{1}{2} \|\mathbf{w}^{jk}\|^2 + C_{jk} \sum_i \xi_i^{jk} \quad (5.5)$$

subject to these constraints:

$$(\mathbf{w}^{jk}) \cdot \varphi(\mathbf{x}_i) + b^{jk} \geq 1 - \xi_i^{jk}, \text{ if } y_i = j$$

$$(\mathbf{w}^{jk}) \cdot \varphi(\mathbf{x}_i) + b^{jk} \leq -1 + \xi_i^{jk}, \text{ if } y_i = k$$

$$\xi_i^{jk} \geq 0$$

In each binary classifier, the regularization parameter and kernel parameter within the method are made to be the same. Then 5-fold cross-validation (CV) is used to tune these parameters so as to achieve a better performance in this classifier. The detailed introduction of this method is given in Section 3.6.2.

The application of SVM consists of the training and classification phases, which handle the training and testing data sets separately. In the training phase, the weights of the decision functions are obtained based on the training set. In the classification phase, the trained neural network classifies the testing data into one of the predefined classes. Sequential minimal optimization (SMO) is applied in this SVM training (Shevade et al., 2000). This algorithm first loops over the entire training set once, and then makes repeated passes over the non-bound samples ($0 < \alpha_i < C_{jk}$) until all of those samples obey the optimal conditions. The algorithm alternates between a single pass over the entire training set and multiple passes over the non-bound subsets until the number of α_i needed to be changed in the entire training set is zero. After all the parameters within this network are decided, the output in this classification task can be predicted. The detailed SMO algorithm is given in Appendix E.2. Note that the SMO algorithm for classification is different from that of regression.

The accuracy of the drawing classifier varies from different drawing classes. Generally, the classification accuracy is around 70%. A little variation may be caused

by the geometric comprehensiveness of the object library. Higher classification accuracy would be achieved with a more comprehensive library.

5.7 3D reconstruction experts

A 3D reconstruction expert has two components: regularity selection and 3D reconstruction formulation.

(1) Regularity selection refers to finding an optimal set of regularities for a drawing dataset. The regularity selection method proposed in Chapter 4 is adopted for each specific drawing domain. A proper regularity set would produce an accurate 3D model in a specific domain. The data used for regularity selection are given in Appendix B. Note that the testing data given in Table 5.2-5.5 is not used for regularity selection.

(2) 3D reconstruction formulation refers to establishing the compliance functions for specific drawing classes. The contributions of the adopted regularities would be reflected in Equation 3.9.

Table 5.1 shows specific regularity sets for the four drawing classes, where the regularities are listed in the order of decreasing rank and their codes are given in Table 4.1. As shown in Table 5.1, the regularities in $S_{\text{class 1}}$ and $S_{\text{class 2}}$ are the same, and the same regularity set can be found between $S_{\text{class 3}}$ and $S_{\text{class 4}}$. The weight coefficient of certain regularity in specific experts is different since the relevance of this regularity to the 3D reconstruction varies in different object classes. Take the weight coefficient of skewed facial symmetry w_{23} as an example. Its value for Class 1 is almost 8 times bigger than that for Class 2. In addition, the value of the coefficient of parallelism w_{21} for Class 3 is almost 5 times bigger than that for Class 4.

Different experts contain different reconstruction models with varying regularity influences. Accordingly, the compliance functions in the form of Equation 3.9, but with specific combination of regularities, are formed to carry out the 3D reconstruction.

Table 5.1 Optimal regularity sets for different reconstruction experts.

Class	Regularities in the order of decreasing rank
1	$S_{\text{class } 1} = \{R4, R9, R3, R1, R2\}$
2	$S_{\text{class } 2} = \{R4, R9, R1, R2, R3\}$
3	$S_{\text{class } 3} = \{R1, R7, R9, R2, R3, R4\}$
4	$S_{\text{class } 4} = \{R9, R7, R2, R3, R4, R1\}$

Moreover, a particular regularity that is not prominent in the categorized classes, such as facial orthogonality for Class 3 and 4, is adopted in the corresponding experts. This is because facial orthogonality exists in the classified object library. It helps to tune the desired shape, though with a minor influence in comparison with the other regularities in the high rank. In this study, the initial estimation model is set to zero, which is a local optimum that needs to be escaped from in the 3D reconstruction process. If a good estimation model is adopted, this regularity may be pruned.

In Chapters 3 and 4, regularity selection and reconstruction formulation are designed for searching a general solution for a whole task. A general object library is developed to cover all possible objects without further classification. The optimality of a regularity set and the compliance function formulation are based upon a global criterion (the minimized reconstruction error in the whole data set). Thus, the solution would entail the loss of accuracy and effectiveness to achieve generality. In the proposed hybrid system, by contrast, they are investigated in the separate sub-regions, which contain objects with special properties. Hence, the solution caters to the possible variation of objects in a local space.

5.8 Procedure of building a hybrid 3D reconstruction system

Before proceeding to the next section, the procedure for building a hybrid intelligent system for 3D reconstruction is summarized as follows:

1. Build an object library to produce the projection drawing data set;
2. Establish a drawing category for general planar objects;
3. Extract 2D features from 2D projection drawings;
4. Build a SVM based multi-classifier based on the drawing data set;
5. Establish reconstruction experts through performing ARD based regularity selection to yield the compliance function for each class.

Note that Step 1 is discussed in Section 3.7, and Steps 2-5 are discussed in Section 5.4-5.7 consecutively.

5.9 Experimental results and discussion from 3D reconstruction

5.9.1 3D reconstruction results obtained from the different reconstruction experts

Based on the architecture of the hybrid system, more sampling data from the library are used to carry out 3D reconstruction. Tables 5.2-5.5 show some of these results.

Table 5.2 Reconstruction result of objects in Class 1 obtained from the reconstruction expert 1.

Case No	Input line drawing	Reconstruction result from different view directions	
1

Chapter 5 A Hybrid Intelligent System for 3D Reconstruction

2
3
4
5

Table 5.3 Reconstruction result of objects in Class 2 obtained from the reconstruction expert 2.

Case No	Input line drawing	Reconstruction result from different view directions	
6
7
8

Chapter 5 A Hybrid Intelligent System for 3D Reconstruction

9
---	---	---	---

Table 5.4 Reconstruction result of objects in Class 3 obtained from the reconstruction expert 3.

Case No	Input line drawing	Reconstruction result from different view directions	
10
11
12
13

Table 5.5 Reconstruction result of objects in Class 4 obtained from the reconstruction expert 4.

Case No	Input line drawing	Reconstruction result from different view directions	
14

15
16
17

5.9.2 3D reconstruction error and time comparison

This section compares the reconstruction performance and time saving between the hybrid system and the single reconstruction system. In the single system, $\{R1, R2, R3, R4, R9\}$ is adopted as the regularity set; and formulated into the compliance function. This single system is used as the benchmark in this study. The reconstruction experts in the hybrid system apply the regularity sets listed in Table 5.1. Note that the weight coefficients of the regularities for the benchmark are different from that of reconstruction experts 1 and 2 in the hybrid system, though the same regularity sets are adopted.

Table 5.6 shows the performance comparison between the two systems. Two indices are used to evaluate the two systems: (1) the reconstruction error E_i and (2) time consumption in the reconstruction process T_{Save} . Their definitions are given in Equation (3.12) and Equation (4.15) separately.

Table 5.6 Reconstruction performance and time saving comparison between the reconstruction experts in the hybrid system and the benchmark.

Class	Object	Reconstruction error		T_{Save}
		Benchmark	Hybrid system	
Class 1	1	0.040	0.045	1%
	2	0.030	0.016	11%
	3	0.125	0.121	6%
	4	0.065	0.057	0%
	5	0.078	0.077	7%
Class 2	6	0.121	0.096	17%
	7	0.086	0.086	7%
	8	0.061	0.059	13%
	9	0.170	0.172	5%
Class 3	10	0.201	0.126	14%
	11	0.093	0.088	6%
	12	0.119	0.081	19%
	13	0.064	0.052	8%
Class 4	14	0.095	0.088	7%
	15	0.128	0.092	8%
	16	0.101	0.040	6%
	17	0.153	0.103	33%

Two observations can be made when analyzing Table 5.6.

(1) The reconstruction errors for objects in Classes 3 and 4 are improved by the hybrid system, but the result is not clear for objects in Classes 1 and 2. Thus, T-test, one of the widely used statistical tests in machine learning (Thomas, 1998), is used to compare the results of Classes 1 and 2 in both systems, as shown in Appendix D.2. The T-test result shows that the reconstruction error of the experts 1 and 2 is equivalent to that of the benchmark. Thus, it can be concluded that the reconstruction accuracy of the whole hybrid system is better than that of the benchmark system. Four 3D reconstruction sub-regions are separately described by the separate reconstruction experts, thus yielding good reconstruction accuracy under an extensive drawing dataset.

(2) As for the time consumption, the reconstruction time consumed by the hybrid system is smaller in almost all the cases compared with the benchmark, except for Object 4, which shows no change in the reconstruction time. The reconstruction algorithms are developed for each specific drawing region, and the time consumption is reduced subsequently.

A further analysis of the above two observations is given as follows. Firstly, the reconstruction experts 3 and 4 yield better reconstruction accuracy in comparison with the benchmark, while the experts 1 and 2 had equivalent reconstruction accuracy. The benchmark applies a reduced regularity set. This exact system itself is obtained from an optimization process (See Chapters 3 and 4 for the optimization process), which is developed from a general object library covering the four classes of this study. Objects from Classes 1 and 2 occupy a high percentage in the library built for the benchmark. Thus, the benchmark is somewhat skewed towards Classes 1 and 2. Secondly, a well-defined reconstruction expert would save more time and improve the reconstruction accuracy. The regularity extraction and the optimization are the two processes that influence the system performances. Since both systems adopt the reduced regularity set, the time consumption for regularity computation is similar. There is no great difference in the computational complexity among the adopted regularity sets. The time difference mainly lies in the optimization process. The experts provide better representations for the four classes than a general one when using the same optimization algorithm. It also improves the reconstruction accuracy because of its optimal formulation. This is manifested by the examples given in Table 5.6. The components in the hybrid system collaborate effectively and produce superior performance.

5.10 Conclusions

There is no general strategy that is always best for reconstructing all objects from all drawings. The more general an approach is, the less successful it will be (Company et al., 2004). Based on this understanding, an intelligent hybrid 3D reconstruction system is developed in this chapter. With this system, feature extraction is utilized to extract valuable information from 2D line drawings, and SVM multi-classification is performed to classify drawings into four classes based on these perceptual features. A reconstruction expert is developed for each class, equipped with the selected regularity set and optimal compliance function formulation, and used to handle the reconstruction. Compared with the benchmark based on a single general reconstruction strategy, the simulation results show that the hybrid system is practical, effective and efficient in 3D reconstruction.

CHAPTER 6

Topological Analysis of a Single Line Drawing

Topological analysis of a line drawing is a preprocessing for 3D reconstruction. Obtaining correct topological information is helpful to effectively identify real face set and decrease the reconstruction complexity. Many issues have been investigated and are reviewed in Section 1.3.3. This chapter discusses some untouched issues in topological analysis of 2D single view line drawings. The methodology to establish manifolds and non-manifolds is discussed. Certain operations are proposed in converting non-manifolds to manifolds when the conversion is possible. A heuristic rule is also proposed for the decomposition of certain complex manifolds to some simple ones.

6.1 Introduction

There are two main objectives to knowing the topology in a line drawing. The first concerns face identification. Two methods are known for identifying the faces of the 2D drawing of an object. The first is an algorithm that can handle general objects, i.e., objects that may be a combination of manifolds and non-manifolds (Shpitalni and Lipson, 1996; Liu and Lee, 2001). But this method has potential flaws in that in certain situations, it cannot identify the faces correctly. For example, circuit (v_5, v_6, v_7, v_8) in Figure 6.1 would be detected as a real face. The other method is applicable for the limited domain of manifold objects only; it identifies a set of properties that relate

vertices, edges and faces in a manifold and use these properties to identify the faces (Liu et al., 2002). This algorithm is very fast compared to the former and it will identify all the faces correctly. Speed is of the essence in any sketch-then-convert interactive system, as the user will not tolerate having to wait for too long. It is preferred that these two algorithms can be selectively and appropriately applied; the first algorithm is applied to drawings of general objects and the latter is used to that of manifolds. Hence it is useful to be able to establish from a drawing whether it depicts a manifold or non-manifold object.

Figure 6.1 The circuit (v_5, v_6, v_7, v_8) is an internal face.

Topological analysis will also facilitate incremental 3D reconstruction. To provide a natural interface for the designers to generate rough solids freely, an online 3D reconstruction system is preferred. A designer may input the design gradually, thus an input drawing is often imperfect or incomplete, yielding manifolds and non-manifolds. An online reconstruction system is illustrated in Figure 6.2, where the reconstruction process is the process of drawing a bit, then reconstructing, then rotating to a new view, freezing the view and then drawing a bit more; and the process repeats. The 3D reconstruction is done simultaneously during or after sketching. Knowledge on the topology of the drawing at any one stage can help in improving the

Figure 6.2 Incremental reconstruction in the online system SRLD:
 (a) frontal geometry; (b) accidental intersection; (c) changing viewpoint;
 (d) adding more details; (e) complete line drawing; (f) reconstructed 3D object.

reconstruction at the next. It allows the “localization” of the modification to the affected topological entities, such as faces and edges. The model constructed should correspond with these sketch changes accordingly in an online 3D reconstruction system.

Topological analysis refers to (1) obtaining the correct face configuration set; (2) detecting the topological properties or changes from a line drawing. Much attention has been paid to the former; little is done with the latter. This chapter focuses on investigating the issues and unsolved problems in topological analysis. It is organized as follows: Section 6.2 sets out the issues and the problems. The representations of non-manifold and manifold line drawings and their related operations are discussed in Section 6.3. The flowchart of the topological analysis is given in Section 6.4, followed by the conclusion in Section 6.5.

6.2 Problem statements

The vertices and edges in a drawing are depicted as real entities of a 3D object by a sketcher. Since a drawing can be viewed as a graph, nodes and arcs in a graph

correspond to real vertices and edges in a 3D object respectively. In this chapter, vertices and edges refer to the entities for both a 3D object and its graph. A detailed definition is given in Section 6.2.1.

6.2.1 Related concepts

Some basic terms that are related with graph theory and topological analysis are given in Section 1.3.3.2. For clarity, more terms used frequently in this chapter are defined below.

General object: General object refers to an object of an unknown type (could be manifold or non-manifold) and it may consist of solids, faces and skeletal structures. The single view line drawing of an object can be viewed as a graph $G = (V, E)$, where all the vertices and edges in a graph correspond to the real vertices of the depicted object.

Face: A face in E^3 is the closure of a nonempty, bounded, and connected subset of E^3 .

A face f in E^2 is the projection of a face in E^3 . Its boundary (denoted by ∂f) is the union of a finite number of line segments.

Vertex: The vertices of a face f , $v(f)$, are defined to be the set of all points for which two line segments, contained in ∂f , can be found whose intersection is the given vertex.

Edge: The edges of a face f , $E(f)$, are defined to be the set of all line segments e , contained in ∂f . The endpoints of e , denoted by $v(e)$, belong to $v(f)$. Any interior points of e , denoted by $i(e)$, does not belongs to $v(f)$. Thus, a vertex is a point in a line drawing that corresponds to a physical point in a general 3D object. There exists $v \in V$ and $e \in E$ for any input drawing.

Sub-graph: A sub-graph of a graph is a component graph that is less than or equal to the original graph. It contains edges and vertices of the original graph.

Disjoint sub-graph: Sub-graphs that have no common vertices are disjoint. Otherwise they are joint sub-graphs.

Degree $d(v)$: The degree of a vertex v is the number of edges incident to v , written as $d(v)$.

Rank $R(e)$: The rank of an edge, denoting the number of faces with boundaries passing through the edge e .

Vertex adjacency matrix G_d : It indicates the connection information of vertices in a line drawing. $G_d = [g_{ij}]_{n \times n}$, $i \neq j$, n is the number of vertices in a drawing; $g_{ij} = 1$ denotes that vertex i and j are connected by an edge, whereas $g_{ij} = 0$ denotes that they are not connected.

Manifold edge: A manifold edge is an edge of an object that is shared exactly by two faces.

Decomposition: In the process of breaking down a graph, the component sub-graphs may or may not be disjoint. If the sub-graphs so obtained are joined, this breakdown is called decomposition; otherwise, it is a partition.

Block: The decomposition of a graph produces two or more sub-graphs each of which is called a block of the decomposition. Furthermore, such a block is a manifold component of the original object.

Connection ratio: The ratio of the actual incident edges to possible incident edges for any vertex of a graph is called its connection ratio (CR). An actual incident edge refers to an edge which is actually connected to a vertex in an input graph, and possible incident edges refer to all the edges that can be connected to a vertex.

6.2.2 Principles

The work in this chapter is carried out based on the following principles:

- **Incremental reconstruction**

The drawing is drawn from a general, non-accidental viewpoint. The on-line sketching system permits changing viewpoint, adding edges to a line drawing and building a 3D model gradually.

- **Undirected vertex-edge graph**

A line drawing is treated as an undirected vertex-edge graph. Furthermore, such a graph may be partitioned into two or more disjoint or joint sub-graphs under certain conditions. As illustrated in Figure 6.3(a), a triangular hole passes through a box; the

Figure 6.3 Line drawings that can be partitioned.

box is treated as one sub-graph, while the hole is another. The line drawing in Figure 6.3(b) is treated as three disconnected sub-graphs corresponding to three simple solids.

The graph in Figure 6.3(c) is decomposed into two joint sub-graphs corresponding to two tetrahedra. A similar partition can be made in Figure 6.3(d) on the two disconnected solids. For each individual sub-graph, face identification and geometric reconstruction are performed separately.

6.2.3 Topological analysis foundations

Two basic rules are formed for the topological analysis of single view line drawings.

Rule 1 For a manifold, the degree of each vertex is at least 3 (a trihedral corner).

In a manifold, the two end vertices of each edge would have at least three incident edges to form two adjacent faces. Thus, the degree of each vertex is at least three. This is the necessary condition of a manifold object and can be used as a criterion to establish a manifold drawing. In the real world, a trihedral polyhedron plays a fundamental role in solid modeling. Machine interpretation of this type of line drawing has been extensively investigated in the last few decades (Sugihara, 1986).

Rule 2 Each edge of a manifold is shared exactly by two faces.

For each edge e , $R(e)=2$.

This is the principle for manifolds (Liu and Lee, 2001). Two real adjacent faces are separated by their common edge. A real edge is determined by the intersection line of two adjacent faces. For non-manifolds, an edge may be shared by one or more real faces, moreover, an edge may be dangling and thus not belong to any face.

6.2.4 Discussions of the line drawing representations

A sketch is a free input by a user. It could be a representation of a manifold, a non-manifold, or a combination of both. In an online reconstruction system, a sketching-based interface enables designers to input the desired shape gradually and

dynamically (Figure 6.2). A sketch is likely to be an imperfect line drawing, as it may be partially finished, or contains redundant or missing lines. Although it is almost impossible to manufacture non-manifolds and they may not represent real objects, there exist theoretical requirements to investigate non-manifolds (Weiler, 1986; Agarwal and Waggenpack, 1992). In addition, some non-manifolds may represent real objects in the real world, such as sheet metal parts. Thus, distinguishing non-manifold and manifold components in a drawing is useful before the 3D reconstruction.

The search for faces from a line drawing is a NP-complete problem (Liu et al., 2002). It is useful to minimize the searching pool to speed up the face identification. The reconstruction complexity is at least n^3 , where n is the number of vertices (Lipson and Shpitalni, 1996). If an input graph can be partitioned into two or more sub-graphs which represent manifold or non-manifold objects, the complexity would be reduced to $n_1^3 + n_2^3 \dots$, where n_1 and n_2 are the number of vertices in the sub-graphs.

Thus, this chapter is devoted to (1) the establishment of the manifolds and non-manifolds from line drawings; (2) the possible conversion of a non-manifold to one or more manifolds; (3) the possible decomposition of a manifold to several blocks. The topological analysis in line drawings needs to be extended to the above issues. Means need to be developed for the detections and partitions within a 2D line drawing.

6.3 Topological analysis of line drawings

Non-manifolds are characterized mathematically by the fact that they possess boundary points whose neighborhoods are not homeomorphic to a planar half space (Requicha, 1977), such as the dangling edge and face shown in Figure 6.4, the non-manifold vertex in Figure 6.5(a), the non-manifold edge in Figure 6.6 and the

overlapping geometric objects in Figure 6.3(d). These non-manifold entities are to be detected in line drawings. Some non-manifolds could be converted to manifolds by some special operations in certain situations.

A non-manifold edge serves as the connection of two or more solids or more than three faces. An edge that is shared by one face or null is a dangling edge. A non-manifold vertex is either shared by two or more solids or a vertex of a non-manifold edge. A redundant edge may also yield a non-manifold object, such as a box with a diagonal edge lying inside. The investigation of non-manifold resulting from redundant edges is not discussed in this study. In addition, non-manifolds such as overlapping geometric objects, which is illustrated in Figure 6.3(d), are not discussed in this chapter since the two disjoint sub-graphs can be handled separately. To facilitate the investigation of non-manifolds in an online system, the definition of an edge is extended to include the case of dangling edges, a special kind of edge that does not belong to any face, such as (v_1, v_7) illustrated in Figure 6.4. Accordingly, v_7 is treated as a special case of vertex in the drawing.

6.3.1 Dangling edge and dangling face detection and deletion

The degree of a manifold vertex is not less than 3. The edges connected to vertices

Figure 6.4 A pyramid with a dangling edge and face (graph G_1).

Chapter 6 Topological Analysis of a Single Line Drawing

of degree 1 are dangling edges. By calculating the degrees of vertices in G , the dangling entities can be identified and removed accordingly. The vertex adjacency matrix G_d , as shown in Table 6.1, is set up for the drawing in Figure 6.4. Vertex v_7 is found with $d(v_7)=1$ from the matrix G_d , thus the edge (v_1, v_7) is detected as a dangling edge. An edge with two end vertices of degree 2 suggests a dangling face, i.e. the edge (v_5, v_6) illustrated in Figure 6.4.

Table 6.1 Vertex adjacency matrix G_d of Figure 6.4

Vertex	1	2	3	4	5	6	7	Degree
1		1	1	1	0	0	1	4
2	1		1	1	0	0	0	3
3	1	1		1	0	1	0	4
4	1	1	1		1	0	0	4
5	0	0	0	1		1	0	2
6	0	0	1	0	1		0	2
7	1	0	0	0	0	0		1

The dangling entities need to be tagged and deleted from G temporally. It means that the removed dangling entities are not involved in further topological analysis, such as in face identification, and will be recovered when the topological analysis process is terminated. Since there is a change in the vertex-edge connection, a further check is performed on the new G_d . Then edge (v_4, v_5) and (v_3, v_6) are found to be dangling edges, and are tagged and deleted accordingly. The check on the vertex-edge connection is terminated until no dangling entities exist in the sub-graph so obtained. Note that when an edge is deleted, its end vertex is retained if this vertex is connected with other vertices in the sub-graph. A potential manifold drawing is obtained since the degree of each vertex in the remaining sub-graph is greater than 2. Furthermore, a sub-graph is handled as a manifold drawing if the degrees of all its vertices are exactly 3. The removed dangling entities and potential manifold so obtained are stored separately as intermediate information. Thus, the online system would only process

the established manifold part and leave the dangling entities unprocessed. A simplified input is produced for the subsequent face identification and 3D reconstruction. If all the edges and vertices in a drawing are subsequently detected as dangling entities and deleted accordingly, the drawing is likely to be an unfinished drawing or of a sheet metal part. The latter is discussed by Lipson (1998) and not addressed in this study.

6.3.2 Detecting non-manifold edge and non-manifold vertex

When the degrees of the vertices in a graph are all greater than 2, it still may represent a non-manifold object, as illustrated in Figure 6.5a and 6.6. A further check on the vertices with degree greater than 3 is needed. Such non-manifolds are likely to contain a non-manifold vertex or edge and can be decomposed into individual manifolds by “breaking” along these non-manifold entities.

Figure 6.5 A non-manifold vertex detection.

6.3.2.1 Non-manifold vertex detection

Let us look at the case where a non-manifold vertex serves as the only connection between or among individual manifolds. Based on rule 1, the degree of such a non-manifold vertex is at least 6, i.e. $d(v_4)=6$ in Figure 6.5(a). In contrast, a manifold

vertex v_4 in Figure 6.5(b) is also found with a high degree $d(v_4)=6$. A specific observation is formed when comparing the two types of vertices.

Observation 1

In a line drawing, there exists a vertex v_a with degree $d(v_a) \geq 6$. Let its adjacent vertices form a set $V_{adjacent}$. If any path that passes through all the vertices in set $V_{adjacent}$ must pass through v_a , then v_a must be a non-manifold vertex.

The vertices adjacent to v_4 form a set $V_{adjacent} (v_1, v_2, v_3, v_5, v_6, v_7)$ in Figures 6.5(a) and 6.5(b) separately. A vertex in $V_{adjacent}$ is selected as a starting vertex to search for a path passing through all the vertices in $V_{adjacent}$. The depth first or breadth first search algorithm can be used (Chartrand and Oellermann, 1993; Gibson, 1985). In Figure 6.5(b), a closed circuit $(v_1, v_2, v_3, v_5, v_6, v_7)$ that passes through all the vertices in $V_{adjacent}$ can be found without passing v_4 . As for Figure 6.5(a), every path must pass through v_4 . Therefore, v_4 in Figure 6.5(a) is a non-manifold vertex.

The degree of a non-manifold vertex is generally higher than that of the other vertices in the same model. Through checking the vertex degree in G_d , the non-manifold vertex is found. A drawing representing a non-manifold is thus established. More observations are proposed in the following sections when there are two or more vertices with high degree.

6.3.2.2 Non-manifold edge detection

For objects with non-manifold edges, the case of two solids meeting along one common edge is studied first. The degree of the end vertices in a non-manifold edge is at least 5 and the number of real faces that pass through this edge must be 4 according to rule 1 and 2 respectively. A non-manifold edge can be identified as follows.

Observation 2

Let (v_a, v_b) be an edge in a line drawing, $d(v_a) \geq 5$, $d(v_b) \geq 5$. All the vertices adjacent

Figure 6.6 A non-manifold object with non-manifold edge.

to the two vertices v_a, v_b form a vertex set $V_{adjacent}$. If any path that passes through all the vertices in $V_{adjacent}$ must pass through v_a or v_b , then (v_a, v_b) must be a non-manifold edge.

The edge (v_7, v_8) in Figure 6.6 is identified as a non-manifold edge based on observation 2.

Although there do not exist two connected solids in Figure 6.7, the edge (v_5, v_6) is

Figure 6.7 Non-manifold edge detection.

identified as a non-manifold edge since no path can be found that passes their adjacent vertices (v_{17} , v_{12} , v_{18} , v_{11} , v_3 , v_8 , v_4 , v_7) without passing vertices v_5 or v_6 . The drawing illustrated in Figure 6.7 needs to be decomposed into two parts through breaking the edge (v_5 , v_6). Two separate sub-graphs (v_{17} , v_{18} , v_5 , v_6 , v_9 , v_{10} , v_{11} , v_{12}) and (v_1 , v_2 , v_3 , v_4 , v_5 , v_6 , v_7 , v_8 , v_{13} , v_{14} , v_{15} , v_{16}) need to be obtained, where the first sub-graph represents a box and the second is in the shape of “L”. The box is null and needs to be subtracted from the solid “L” in the reconstruction. The decomposition method is discussed next.

6.3.3 Non-manifold decomposition

A graph decomposition algorithm is used to cluster vertices in a non-manifold drawing and identify manifold blocks. In a traditional vertex classification or organization problem, vertices with some common features such as distance between nodes, flow density and so on, are grouped together. However, a line drawing does not have these features since all the vertices are connected by the true edges of the object it depicts. The information that can be used to cluster vertices is therefore quite limited. The adopted graph decomposition method is first proposed by Owen (1970)

Figure 6.8 A non-manifold object partition along the non-manifold edge.

Chapter 6 Topological Analysis of a Single Line Drawing

and later revised by Chen et al. (2002). It is used in a non-manifold line drawing in which, it is assumed, a non-manifold vertex or edge serves as the only connection between two manifolds.

Firstly, an analogous self-organizing device, derived from the inherent structure of an input graph, is defined. This device gives a weight to every edge. The value of the weight is calculated from the connections between vertices. It determines an initial sub-graph structure. Then, an ordered search device is set up to place the vertex in order. An appropriate vertex order would reduce the search time and improve the probability of finding an optimal partition. The degrees of non-manifold vertices are usually greater than those manifold vertices. Hence they are placed in the front of the sorting array and chosen to start a new sub-graph. To establish a flexible partition, an average connection ratio is given as the initial value to control the partition process, for example the experimental value for Figure 6.8 is 0.68. This parameter can be modified to obtain desired partition result, if needed. Finally, a corrective device is adopted to modify and check the elements in a sub-graph based on the two constraints: (1) the degree of each vertex in each sub-graph is not less than 3; (2) the common vertices of the joint sub-graphs are limited to the non-manifold vertices. The decomposition process is heuristic. A sub-graph is required to be as large as possible. By defining the decomposition rules in this way, control can be exercised over the size of sub-graphs and the degree of connectedness within them. A detailed introduction of the adopted decomposition algorithm is given in Appendix G.

This decomposition is somewhat arbitrary for some complex objects, such as a drawing with more than three manifolds. Good performance is obtained in testing the connected pyramidal objects. Though prior knowledge of the non-manifold edges and vertices are helpful for the decomposition, they are not necessary. When the given

non-manifold vertices are not correct, the decomposition algorithm may still provide a desired decomposition result. This is one of the reasons of selecting the revised Owen's algorithm. The other reason to adopt this method is that no prior constraints are needed, such as the number of sub-graphs, the size of sub-graphs and so on. These constraints are required as a prior in the traditional graph partition algorithms, though they are not known before the decomposition. In this sense, the decomposition problem here is not a standard graph decomposition problem. Future research is recommended to investigate special non-manifolds in particular applications to achieve desired results.

6.3.4 Possible manifold decomposition and internal face identification

The face identification method (Liu and Lee, 2001) may yield some internal faces whose circuits do not represent actual faces, such as the circuit (v_5, v_6, v_7, v_8) in Figure 6.1. The invalid face needs to be deleted before the reconstruction process to decrease the reconstruction complexity. Observation 3 is used to identify an invalid face.

Observation 3

If there exists an edge set $E = \{e_1, e_2, \dots, e_n\}$, with $R(e_i) > 2$ and $n \geq 3$, where n is the number of edges in E , an anomaly would occur for a manifold drawing. A circuit must be an invalid face if the degrees of all its edges are greater than 2.

Based on this observation, the decomposition of a manifold drawing is possible when an internal face shared by two or more blocks is found. Take the drawing in Figure 6.9 as an example. Circuit (v_5, v_6, v_7, v_8) is an internal face; its vertices can be used to decompose the graph into two sub-graphs, $(v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)$ and $(v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12})$.

Chapter 6 Topological Analysis of a Single Line Drawing

The vertices of an internal face are used as cut vertices to collapse a graph. A vertex set $V_{internal}$ is formed. A multilevel graph decomposition method is adopted to decompose a manifold drawing. It overcomes the shortcomings of the standard graph partition algorithm and provides good partitions at lower cost for practical problems (Hendrickson and Leland, 1995). In the partition scheme, the size of a graph is firstly reduced by removing cut vertices and the edges between any two cut vertices. The original graph is thereby coarsened and several disjoint sub-graphs are produced. This is a partition. The number of the sub-graphs indicates the number of blocks constituting the manifold. Then, an uncoarsening process is performed for each individual sub-graph to yield graph decomposition. Each partitioned sub-graph is uncoarsened through recovering the removed vertices according to the vertex connections of the original graph. In the uncoarsening process, the disjoint sub-graphs are converted to the joint sub-graphs, whose common vertices belong to $V_{internal}$. A manifold decomposition is thus obtained. Take Figure 6.9(b) as an example. The two procedures are explained as follows.

Firstly, the vertices in $V_{internal}$ are removed from the original graph as illustrated in Figure 6.9(b). Note that the removal of a cut vertex is to delete this vertex and break all its incident edges from the original graph. The vertex adjacent to a cut vertex is retained if it does not belong to $V_{internal}$, meanwhile, the edges between the vertices in $V_{internal}$ are all removed. Thus, two disjoint sub-graphs G_1 and G_2 are obtained, whose vertices are (v_1, v_2, v_3, v_4) and $(v_9, v_{10}, v_{11}, v_{12})$ separately. This means that the input drawing can be decomposed into two blocks.

Secondly, a vertex recovery procedure is performed for each vertex in $V_{internal}$ to form the joint sub-graphs. The vertex recovery is to copy and paste a vertex from $V_{internal}$ to a sub-graph G_i based on the connection information of the original graph.

Chapter 6 Topological Analysis of a Single Line Drawing

When a vertex is added to G_i , it does not mean all its incident edges are added automatically, only the edge that is between this vertex and the vertices in G_i are added. If there is no connection among all the vertices in G_i and the added vertex, this vertex adding process is terminated and the other vertices in $V_{internal}$ are tried. If no vertex in $V_{internal}$ can be added to G_i , the recovery process stops. This uncoarsening

Figure 6.9 The decomposition of a manifold line drawing:

(a) a manifold drawing containing two blocks; (b) the input graph is collapsed into two disjoint sub-graphs through removing the vertices in the internal face; (c) vertices recovered for the sub-graph representing a frustum; (d) a frustum obtained with the edges recovered among the added vertices; (e) vertices recovered for the sub-graph representing a rectangular block; (f) a rectangular block with the edges recovered among the added vertices.

process repeats until all the disjoint sub-graphs are processed. Two sub-graphs are produced as illustrated in Figure 6.9 (c) and (e). The missing edges between the cut vertices in each sub-graph are recovered based on the original graph. The joint sub-graphs are obtained, as illustrated in Figure 6.9(d) and (f). Two manifold blocks are obtained, where the vertex degree in each sub-graph is not less than 3.

This graph decomposition algorithm is based on the prior identification of cut vertices. It is easy to understand and implement. As for the algorithm discussed in Section 6.3.3, the decomposition can be done without knowing the cut vertex, but the process is more complex. Thus, the two algorithms have their own advantages and drawbacks. The graph uncoarsening algorithm can also be used for non-manifolds, given that the cut vertices are known correctly.

6.4 Flowchart and discussions of the topological analysis

A suitable face identification method needs to be selected to yield correct faces. Figure 6.10 illustrates the flowchart of the topological analysis of a single view line drawing. The input graph is analyzed first to establish whether it depicts a manifold or non-manifold object. The topological analysis can be summarized as:

- (1) Find and store possible dangling entities in a line drawing;
- (2) Find possible disjoint neighborhood entities and decompose the input graph;
- (3) Choose an appropriate face identification method to obtain the actual face set;
- (4) Use internal face to decompose certain complex manifolds.

Step 1 is discussed in Section 6.3.1. Sep 2 is discussed in Section 6.3.2. Step 4 is discussed in Section 6.3.4. The face identification method in Step 3 is fully exploited in Liu and Lee (2001) and Liu et al. (2002). The method selection is discussed as follows.

Chapter 6 Topological Analysis of a Single Line Drawing

With the current investigation, a line drawing is handled as a manifold if the degree of each vertex in a drawing is exactly 3 after all the dangling entities are removed. A face identification method developed for manifold drawing can be adopted (Liu and Lee, 2001). As for a drawing, in which at least one of its vertex degrees is found to be greater than 3, Observations 1 and 2 would be used to identify some special non-manifold cases, such as non-manifold vertex or edge. However, such identification may not account for all the non-manifolds. The proposed observation serves to be the

Figure 6. 10 Flowchart of the topological analysis of a single line drawing.

necessary condition. In such cases, a general face identification method (Liu and Lee, 2001), which can handle manifolds and non-manifolds, would be used because the topological properties of a drawing is not established.

Furthermore, for a non-manifold drawing, decomposition is performed to obtain a manifold drawing in some limited conditions. Non-manifold vertex and edge need to be detected first. They serve as the breaking points to produce manifold sub-graphs.

As for an established manifold drawing, the decomposition of some complex manifolds is possible through the detection of the internal faces. The decomposition result may provide some cues to building a 3D object block by block for incremental 3D reconstruction.

6.5 Conclusions

This chapter discusses some basic problems for the topological analysis of a single line drawing. Efficient and effective face identification would be realized if the topological properties of a drawing can be established. According to the current investigations, manifold and non-manifold drawings cannot be established definitely. A drawing is identified to be manifold if the degree of each vertex is exactly 3; a non-manifold drawing is detected when at least one of its vertices is found with degree less than 3. The observations proposed for detecting non-manifold edges and vertices are only necessary conditions, and they may not distinguish non-manifolds from manifolds definitely. Therefore, much work is needed for the establishment of non-manifolds and manifolds from a single view line drawing alone. Future work need to be focused on investigating topological properties of a line drawing, developing effective decomposition algorithms and incorporating the solutions to an on-line 3D reconstruction system.

CHAPTER 7

Conclusions and Recommendations

This chapter draws conclusions from the results presented in the preceding chapters and makes recommendations for future work.

7.1 Conclusions

Many issues and unsolved problems exist in the 3D reconstruction from a single view line drawing. This thesis proposes a set of methods to yield desirable results that best correspond to human perception. Advanced artificial intelligence techniques, such as support vector machine, feature selection etc., are applied to address issues in the 3D reconstruction. With the proposed methods, the optimal formulation for compliance function, the effective regularity extraction and selection, and the hybrid intelligent system are investigated.

The compliance function is the specific characterization of the conversion of a line drawing to a 3D object. A good formulation is important for the 3D reconstruction. A systematic method is proposed to address the issues in formulating the compliance function. Issues such as the best form for the function, the overlapping and conflicting information in the regularities in the formulation, and computing for the coefficients of the regularity terms in the function, are investigated with the use of support vector regression estimation.

Chapter 7 Conclusions and Recommendations

Various regularities can be extracted for 3D reconstruction, but not all the regularities are equally deployable: certain regularities may correspond mainly to noise, not information; some may overlap with each other or are not too relevant to the reconstruction. The automatic relevance determination approach coupled with the Bayesian framework and support vector regression estimation is used to prune the irrelevant regularities and retain the relevant ones. The proposed method not only reduces the computational complexity of the 3D reconstruction by establishing a reduced regularity set, but also provides a robust and feasible regularity recognition method.

The more general a 3D reconstruction approach is, the less successful it will be. The existing systems are single-track, containing one general solution for all drawings. Such a general system may not be able to produce the best solution for all possible drawings. A hybrid intelligent 3D reconstruction system is proposed, in which line drawings are classified with the technique of support vector machine multi-classification, and reconstruction experts are developed for each class, equipped with the selected regularity set and optimal compliance function formulation. The proposed system can improve the reconstruction accuracy and efficiency through selectively adopting suitable experts for each specific reconstruction.

Finally, this thesis presents a framework to topologically identify the manifold and non-manifold elements from a line drawing. Since a 3D reconstruction system is expected to have automatic processing ability, the conversion of non-manifold line drawings to manifolds is investigated. In addition, the possible partition of a manifold or non-manifold line drawing into several blocks is also investigated.

7.2 Recommendations for future work

Future work is recommended for the extension of the current 3D reconstruction system with a more natural and powerful reconstruction function. Several issues need to be resolved.

(1) Combining the gesture-based reconstruction system with the line drawing-based system is meaningful to capture the design ideas quickly and flexibly. Through exploiting the benefits from both systems, an active and natural sketching interface can be established. Furthermore, 3D objects can be built quickly and effectively for practical applications. Take a box with one cut corner as an example. A sketcher may draw the box first and then scribble over the unwanted part. A gesture that encodes “deletion” needs to be predefined and included in a 3D reconstruction system. Some other gestures are also needed, such as copying, cutting and pasting. Introducing gestures to a line drawing-based 3D reconstruction system would yield easy manipulation and powerful modeling technique for sketching-based design.

(2) An incremental interpretation and reconstruction process on the level of block are needed for practical design. A block is termed feature in solid modeling, referring to a protrusion, a depression, a hole and so on. If these features can be recognized, some 3D models can be constructed block-by-block from a line drawing. Automatic feature recognition has been an active research area in solid modeling (Kyprianou, 1980). It provides meaningful cues for mechanical design and fabrication. Such recognition will provide an effective function for sketching-based design and facilitate the integration of CAD/CAPP/CAM systems. It is suggested to integrate the graph algorithms with some heuristic rules to recognize blocks from a 2D drawing alone.

(3) There lacks adequate measures to integrate the interpretation and geometric reconstruction process effectively and thoroughly. The two processes are investigated

Chapter 7 Conclusions and Recommendations

separately in the existing reconstruction systems. The former is to identify the real faces from a drawing, while the latter is to construct the 3D object. The information contained in a drawing is not interpreted sufficiently and conveyed adequately to the reconstruction process. Some important cues, such as the constitutions of blocks in an object and their relative positions, are ignored, though they can be extracted and used. This neglect can produce undesirable or unexpected reconstruction results. Take Figure 2.7 (b) as an example, each reconstructed block corresponds to human perception, while the whole result does not. It is necessary for the computer to predict and overcome the undesirable result through some automatic measures. The constraints on how the different blocks are combined together should be detected and conveyed in the reconstruction process. They should be inherited and enforced in the individual block reconstructions. For example, the constraints would enable the transformation of the related vertex coordinates from one block to its neighbor for the subsequent reconstruction. With the smaller number of vertices in each block, the complexity of 3D reconstruction would be decreased.

(4) Pattern recognition is needed to establish the projection types of an input drawing. This would help to form an initial 3D model estimation efficiently and effectively. The line labeling scheme can also be added in future estimation procedure, for two reasons. Firstly, it has proven effective in the reconstruction of drawings with hidden topology (Varley and Martin, 2000a). Secondly, it seems useful in the system proposed by Company et al. (2004). Furthermore, different estimation methods should be investigated and embedded in different reconstruction experts.

A revised hybrid intelligent system can be constructed as illustrated in Figure 7.1. In each classified drawing region, different support vector machines (or other neural networks, such as Markov network, Hopfield network or Bayesian network, which

Figure 7.1 A revised hybrid intelligent system for 3D reconstruction.

have been proven useful in image processing) compete until the most competent one is found to tackle 3D reconstruction.

(5) An input drawing is very likely to be inaccurate since a designer would sketch roughly on a piece of paper. The current investigation mainly focuses on the real projection of an object (Qin et al., 2000; Varley and Martin, 2000a; Company et al., 2004). Thus, properly handling inaccurate drawing still remains an issue. This thesis focuses on the accurate drawings; further investigation is needed to extend the current research to the inaccurate user-produced drawings. Sketching style or habit of a designer need to be identified and considered. An input sketch needs to be properly interpreted and recognized for 3D reconstruction.

References

Adgar, A., Cox, C., Emmanouilidis, C., MacIntyre, J., Mattison, P., McGarry, K., Oatley, G. and Taylor, O. The application of adaptive systems in condition monitoring, *International Journal of Condition Monitoring and Diagnostic Engineering Management*, Vol. 1. No. 1, pp.13-17, 1998.

Agarwal, S.C. and Waggenspack, J.W.N. Decomposition method for extracting face topologies from wire frame models, *Computer Aided Design*, Vol. 24, No. 3 pp.123-140, 1992.

Amo, P., Ferreras, F.L., Cruz, F. and Rosa, M. Smoothing functions for automatic relevance feedback in information retrieval, in *Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA'00)*, pp. 115-119, 2000.

Anne, R., Guido, W. and Winfried, H. Effort-saving product representations in design-results of a questionnaire survey, *Design Studies*, Vol. 22, No. 6, pp. 473-491, 2001.

Arnheim, R. *Visual Thinking*, University of California Press, Berkeley, 1969.

Bagali, S. and Waggenspack, J.W.N. A shortest path approach to wireframe to solid model conversion, in *Proceedings of Third symposium on Solid Modeling and Applications*, Salt Lake City, US, pp. 339-349, 1995.

Baraldi, A. and Blonda, P. A survey of fuzzy clustering algorithms for pattern recognition—part II, *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, Vol. 29, No. 6, pp. 786-801, 1999.

Barhak, J. and Fischer, A. Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 7, No. 1, pp. 1-16, 2001.

Barzilay, O. and Brailovsky, V.L. On domain knowledge and feature selection using a support vector machine, *Pattern Recognition Letters*, Vol. 20, No. 5, pp. 475-484, 1999.

Belue, L.M. and Bauer, Jr.K.W. Determining input features for multi-layer perceptrons, *Neurocomputing*, Vol. 7, No. 2, pp. 111-121, 1995.

Bergan, R.C., and Trubatch, S.L. Fuzzy logic and hybrid approaches to web intelligence gathering and information management, in *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'02)*, Vol. 2, pp. 1033-1038, 2002.

References

- Bishop, C.M. *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.
- Bloomenthal, M., Zeleznik, R.C., Fish, R., Holden, L.S., Forsberg, A.S., Riesenfeld, R., Cutts, M., Drake, S., Fuchs, H. and Cohen, E. Sketch-N-Make: automated machining of CAD sketches, in *Proceedings of the 1998 ASME 8th Computers In Engineering Conference*, Atlanta, US, pp. 1-11, 1998.
- Bonissone, P. Hybrid soft computing systems: where are we going? in *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*, Berlin, Germany, pp. 739-746, August 2000.
- Bradley, P.S. and Mangasarian, O.L. Feature selection via concave minimization and support vector machines, in *Proceedings of the 15th International Conference on Machine Learning (ICML-98)*, Madison, WI, US, pp.82-90, 1998.
- Braid, L. and Wang, P. Three-dimensional object recognition using gradient descent and the universal three-dimensional array grammar, in *Proceedings of SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, Boston, MA, Vol. 1607, pp. 711-718, 1991.
- Brake, G.M.T., Karssemeijer, N. and Hendriks, J.H.C.L. An automatic method to discriminate malignant masses from normal tissue in digital mammograms, *Journal of Physics in Medicine and Biology*, Vol. 45, No.10, pp. 2843-2857, 2000.
- Brill, F.Z., Brown, D.E. and Martin, W.N. Fast genetic selection of features for neural network classifiers, *IEEE Transactions on Neural Networks*, Vol. 3, No. 2, pp. 324-328, 1992.
- Burges, C.J.C. A tutorial on support vector machines for pattern recognition, *Data Mining Knowledge Discovery*, Vol. 2, No. 2, pp. 955-975, 1998.
- Cenci, M., Nagar, C. and Vecchione, A. PAPNET-assisted primary screening of conventional cervical smears, *Anticancer Research*, Vol. 20, No. 5C, pp. 3887-3889, 2000.
- Chakravarty, I. A generalized line and junction labeling scheme with applications to scene analysis, *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 1, No. 2, pp. 202-205, 1979.
- Chao, K., Chen, Y. R., Early, H. and Park, B. Color image classification systems for poultry viscera inspection, *Applied Engineering in Agriculture*, Vol. 15, No. 4, pp. 363-369, 1999.
- Chartrand, G. and Oellermann, O.R. *Applied and Algorithmic Graph Theory*, McGraw-Hill, New York, 1993.
- Chen, C.H. and Occña, L.G. Knowledge decomposition for a product design blackboard expert system, *Artificial Intelligence in Engineering*, Vol. 14, No.1, pp. 71-82, 2000.
- Chen, C.H., Wu T. and Occña L.G. Knowledge organization of product design blackboard systems via graph decomposition, *Knowledge-based Systems*, Vol. 15, No. 7, pp. 423-435, 2002.

References

- Chen, Y.Q., Zhou, X.S. and Huang, T.S. One-class SVM for learning in image retrieval, in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 34-37, 2001.
- Chenoweth, T. and Obradovic, Z. An explicit feature selection strategy for predictive models of the S&P 500 Index, *Neurovest Journal*, Vol. 3, No. 6, pp. 14-21, 1995.
- Cho, S. and Chow, T.W.S. Neural computation approach for developing a 3-D shape reconstruction model, *IEEE Transactions on Neural Networks*, Vol. 12, No. 5, pp. 1204-1214, 2001.
- Chu, W., Keerthi, S. S. and Ong, C.J. Bayesian trigonometric support vector classifier, *Neural Computation*, Vol. 15, No. 9, pp. 2227-2254, 2003.
- Clowes, M.B. On Seeing Things, *Artificial Intelligence*, Vol. 2, No. 1, pp. 79-112, 1971.
- Company, P., Contero, M., Conesac, J. and Piquer, A. An optimization-based reconstruction engine for 3D modeling by sketching, *Computers and Graphics*, Vol. 28, No. 6, pp. 955-979, 2004.
- Contero, M., Naya, F., Gomis, J.M. and Conesa, J. Calligraphic interfaces and geometric reconstruction, in *Proceedings of 12th ADM International Conference on Design Tools and Methods in Industrial Engineering*, Rimini, Italia, pp. 110-110, September 2001.
- Cooper, M.C. Linear-time algorithms for testing the realisability of line drawings of curved objects, *Artificial Intelligence*, Vol. 108, No. 1-2, pp. 31-67, 1999.
- Cooper, M.C. Linear constraints for the interpretation of line drawings of curved objects, *Artificial Intelligence*, Vol. 119, No. 1-2, pp. 235-258, 2000.
- Cortes, C. and Vapnik, V.N. Support vector networks, *Machine Learning*, Vol. 20, No. 3, pp. 273- 297, 1995.
- Courter, S.M. and Brewer, J.A. Automated conversion of curvilinear wireframe models to surface boundary models: a topological approach, *Computer Graphics*, Vol. 20, No. 4, pp. 171-178, 1986.
- Dash, M. and Liu, H. Feature selection for classification, *Intelligent Data Analysis*, Vol. 1, No. 3, pp. 374-385, 1997.
- Dijk, C.G.C. *Interactive Modeling of Transfinite Surfaces with Sketched Design Curves*, PhD. Dissertation, Delfy University Press, 1994.
- Dougherty, M.S. and Cobbett, M.R. Short-term inter-urban traffic forecasts using neural networks, *International Journal of Forecasting*, Vol. 13, No. 1, pp. 21-31. 1997.
- Draper, S.W. *Reasoning about Depth in Line-Drawing Interpretation*, PhD. Dissertation, Sussex University, 1980.
- Draper, S.W. The use of gradient and dual space in line-drawing interpretation, *Artificial Intelligence*, Vol. 17, No. 1, pp. 461-508, 1981.

References

- Duan, K., Keerthi, S.S. and Poo, A.N. Evaluation of simple performance measures for tuning SVM hyperparameters, *Neurocomputing*, Vol. 51, pp. 41-59, 2003a.
- Duan, K., Keerthi, S.S., Shevade, S.K., Chu, W. and Poo, A.N. Multi-category classification by soft-max combination of binary classifiers, *Fourth International Workshop on Multiple Classifier Systems*, Surrey, UK, pp. 125-134, 2003b.
- Dutton, R.D. and Brigham, R.C. Efficiently identifying the faces of a solid, *Computers and Graphics in Mechanical Engineering*, Vol. 7, No. 2, pp. 143-147, 1983.
- Eggli, L., Brüderlin, B.P. and Elber, G. Sketching as a solid modeler tool, *Third Symposium on Solid Modeling and Applications*, ACM SIGGRAPH, 1995, pp. 313-321.
- Egmont-Petersen, M., de Ridder, D. and Handels, H. Image processing with neural networks - a review, *Pattern Recognition*, Vol. 35, No. 10, pp. 2279-2301, 2002.
- Frangi, A.F., Niessen, W.J. and Viergever, M.A. Three-dimensional modeling for functional analysis of cardiac images: a review, *IEEE Transactions on Medical Imaging*, Vol. 20, No. 1, pp. 2-25, 2001.
- Friedberg, S.A. Finding axes of skewed symmetry, *Computer Vision, Graphics, and Image Processing*, Vol. 34, No. 2, pp.138-155, 1986.
- Ganter, M.A. and Uicker, J.J. From wireframe to solid geometric: Automated conversion of data representations, *Computers in Mechanical Engineering*, Vol. 2, No. 2, pp. 40-45, 1983.
- Gibbson, A. *Algorithmic Graph Theory*, Cambridge University Press, Cambridge, 1985.
- Goebel, K. and Yan, W. *Feature Selection for Tool Wear Diagnosis Using Soft Computing Techniques*, Technical Report 2001CRD011, GE Research & Development Center, 2001.
- Goldschmidt, G. On visual design thinking, *Design Studies*, Vol. 15, No.2, pp. 158-175, 1994.
- Grimstead, I.J. and Martin, R.R. Creating solid models from single 2D sketches, in *Proceedings of Third Symposium on Solid Modeling and Applications*, Hoffmann, C. and Rossignac, J. (Eds), Salt Lake City, US, ACM Press, pp. 323-337, 1995.
- Grimstead, I.J. *Interactive Sketch Input of Boundary Representation Solid Models*, PhD. Dissertation, Cardiff University of Wales College, 1997.
- Gu, P. and Yuan, X. Neural network approach to the reconstruction of freeform surfaces for reverse engineering, *Journal of Computer Aided Design*, Vol. 27, No. 1, pp. 59-69, 1995.
- Guo, G.D., Jain, A.K., Ma, W.Y. and Zhang, H.J. Learning similarity measure for natural image retrieval with relevance feedback, *IEEE Transactions on Neural Networks*, Vol. 13, No. 4, pp. 811-820, 2002.

References

- Hanrahan, P.M. Creating volume models from edge vertex graphs, *Computer Graphics, ACM SIGGRAPH Proceedings*, Vol. 16, No. 3, pp. 77-84, 1982.
- Hastie, T., Tibshirani, R. and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Berlin, Springer, 2001.
- Haykin, S. *Neural Networks: A Comprehensive Foundation* (2nd ed.), Prentice Hall, New Jersey, 1999.
- Henderson, M.R., Srinath, G., Stage, R. Walker, K. and Regli, W. Boundary representation-based feature identification, in *Advances in Feature Based Manufacturing*, Chapter 1, Shah, J.J., Mäntylä, M. and Nau, D.S. (Eds), Elsevier Science Ltd, pp. 15-38, 1994.
- Hendrickson, B. and Leland, R. A multilevel algorithm for partitioning graphs, in *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM) - Volume 00*, ACM Press, New York, US, 1995.
- Hill, J.F.S. *Computer Graphics*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.
- Hoffmann, M. and Várady, L. Free-form surfaces for scattered data by neural networks, *Journal for Geometry and Graphics*, Vol. 2, No. 1, pp. 1-6, 1998.
- Hojnicki, J.S. and White, P.R. Converting CAD wireframe data to surface representations, *Computers in Mechanical Engineering*, , pp. 19-25, March/April 1988.
- Hong, P.Y., Tian, Q. and Huang, T. S. Incorporate support vector machines to content-based image retrieval with relevance feedback, in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 750-753, 2000.
- Hsu, C.W. and Lin, C. J. A comparison of methods for multi-class support vector machines, *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, pp. 415-425, 2002.
- Huffman, D.A. Impossible objects as nonsense sentences, *Machine Intelligence*, Vol. 6, Meltzer, B. and Michie, D. (Eds.), Edinburgh University Press, Edinburgh, pp. 295-323, 1971.
- Huffman, D.A. A duality concept for analysis of polyhedral scenes, *Machine Intelligence*, Vol. 8, Elcock, E.W. and Michie, D. (Eds.), Ellis Horwood, pp. 475-492, 1977.
- Igarashi, T., Matsuoka, S. and Tanaka, H. Teddy: a sketching interface for 3D freeform design, in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM SIGGRAPH 99, ACM Press/Addison-Wesley Publishing Co., pp. 409-416, 1999.
- Jacobs, R., Jordan, M., Nowlan, S. and Hinton, G. Adaptive mixtures of local experts, *Neural Computation*, Vol. 3, No.1, pp.79-87, 1991.
- Jain, A. K. and Zongker, D. Feature-selection: evaluation, application, and small sample performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 2, pp. 152-157, 1997.

References

- Jebara, T. and Jaakkola, T. Feature selection and dualities in maximum entropy discrimination, in *Proceeding of the 16th Conference on Uncertainty in Artificial Intelligence*, Stanford University, CA, US, pp. 291-300, 2000.
- Jenkins, D.L. and Martin, R.R. Applying constraints to enforce users' intentions in free-hand 2-D sketches, *Intelligent Systems Engineering*, Vol. 1, No. 1, pp. 31-49, 1992.
- Jezekiel, B. and Dibyendu, N. A neural network approach for reconstructing surface shape from shading, in *Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP-98)*, Vol. 2, Chicago, US, pp. 972-976, 1998.
- Joachims, T. *The Maximum-Margin Approach to Learning Text Classifiers: Method, Theory and Algorithms*, PhD. Dissertation, University of Dortmund, 2000.
- Jordan, M. A statistical approach to decision tree modeling, in *Proceedings of the Seventh ACM Conference on Computational Learning Theory*, pp. 13-20, New Brunswick, NJ, July 1994.
- Kanade, T. Recovery of the three-dimensional shape of an object from a single view, *Artificial Intelligence*, Vol. 17, No. 1, pp. 409-460, 1981.
- Kanatani, K. Group-theoretical methods in image understanding, *Springer Series in Information Sciences*, No. 20, Springer-Verlag, 1990.
- Keerthi, S.S. and Gilbert, E.G. Convergence of a generalized SMO algorithm for SVM classifier design, *Machine Learning*, Vol. 46, No. 1-3, pp. 351-360, 2002.
- Keerthi, S.S. and Shevade, S.K. SMO algorithm for least-squares SVM formulations *Neural Computation*, Vol. 15, No. 2, pp. 487-507, 2003.
- Kewley, R.H., Embrechts, J.J. and Breneman, C. Data strip mining for the virtual design of pharmaceuticals with neural networks, *IEEE Transactions on Neural Networks*, Vol. 11, No. 3, pp. 668-679, 2000.
- Kirousis, L.M. and Papadimitriou, C.H. The complexity of recognizing polyhedral scenes, *Journal of Computer Systems Science*, Vol. 37, No. 1, pp. 14-38, 1988.
- Kohonen, T. *Self-organization and Associative Memory*, Springer, New York, 1989.
- Koopmans, L.H. *Introduction to Contemporary Statistics Methods* (2nd ed.), PWS Publishers, Boston, Mass, 1987.
- Kuhn, H.W. and Tucker, A.W. Nonlinear programming, in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Neyman, J. (ed.) Berkeley, University of California Press, pp. 481-492, 1951.
- Kuncheva, L.I. and Jain, L.C., Nearest neighbor classifier: simultaneous editing and feature selection, *Pattern Recognition Letters*, Vol. 20, No. 11-13, pp. 1149-1156, 1999.
- Kwok, J.T. Linear Dependency between ε and the Input Noise in ε -Support Vector Regression, in *Proceedings of International Conference on Artificial Neural Networks (ICANN)*, Dorffner, G., Bishof, H. and Hornik, K. (Eds), pp. 405-410, 2001.

References

- Kwok, T.Y. Support vector mixture for classification and regression problems, in *Proceedings of the 14th International Conference on Pattern Recognition (ICPR'98)*, Brisbane, Australia, 1998, pp. 255-258.
- Kyprianou, L.K. *Shape Classification in Computer Aided Design*, PhD. Dissertation, Kings College, University of Cambridge, 1980.
- Lamb, D. and Bandopadhyay, A. Interpreting a 3D object from a rough 2D line drawing, in *Proceeding of IEEE Visualization'90*, pp. 59-66, October 1990.
- Langbein, F.C., Mills, B.I., Marshall, A.D. and Martin, R.R. Recognizing geometric patterns for beautification of reconstructed solid models, in *Proceedings of International Conference on Shape Modeling and Applications*, Italy, pp. 10-19, 2001.
- Leclerc, Y.G. and Fischler, M.A. An optimization based approach to the interpretation of single line drawings as 3D wire frames, *International Journal of Computer Vision*, Vol. 9, No. 2, pp. 113-136, 1992.
- Lee, S.J., Haralick, R.M. and Zhang, M.C. Understanding objects with curved surfaces from a single perspective view of boundaries, *Artificial Intelligence*, Vol. 26, pp. 145-169, 1985.
- Lee, S., Lo, C., Wang, C., Chung, P., Chang, C., Yang, C. and Hsu, P. A computer-aided design mammography screening system for detection and classification of microcalcifications, *International Journal of Medical Informatics*, Vol. 60, No. 1, pp. 29-57, 2000.
- Lim, W.P. and Siti, M.S. 3D object reconstruction and representation using neural networks, in *Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques (GRAPHITE 2004)*, Singapore, pp. 139-147, 2004.
- Lin, S.H., Kung, S.Y. and Lin, L.J. Face recognition/detection by probabilistic decision based neural network, *IEEE Transactions on Neural Networks*, Vol. 8, No. 1, pp. 114-132, 1997.
- Lipson, H. and Shpitalni, M. A new interface for conceptual design based on object reconstruction from a single freehand sketch, *Annals of the CIRP*, Vol. 44, No. 1, pp. 133-136, 1995.
- Lipson, H. and Shpitalni, M. Optimization-based reconstruction of a 3D object from a single freehand line drawing, *Journal of Computer Aided Design*, Vol. 28, No. 8, pp. 651-663, 1996.
- Lipson, H. *Computer Aided 3D Sketching for Conceptual Design*, Ph.D. Dissertation, Israel Institute of Technology, 1998.
- Liu, C. and Wechsler, H. Evolutionary pursuit and its application to face recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 6, pp. 570-582, 2000.
- Liu, J.Z. and Lee, Y.T. *A Tool for 3D Surface Reconstruction from a 2D Line Drawing (SRLD)*, Technical Report, School of Mechanical & Production Engineering, Nanyang Technological University, Singapore, 2000.

References

- Liu, J.Z. and Lee, Y.T. A graph-based method for face identification from a single 2D line drawing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 10, pp. 1106-1119, 2001.
- Liu, J.Z., Lee, Y.T. and Cham, W.K. Identifying faces in a 2D line drawing representing a manifold object, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 12, pp. 1579-1593, 2002.
- Liu, J.Z. and Tang, X. A review of methods in face identification from line drawings for 3D object reconstruction, Invited paper in *Proceedings of International Symposium on Computer Vision, Object Tracking and Recognition*, Beijing, China, pp. 1-8, 2004a.
- Liu, J.Z. and Tang, X. Efficient search of faces from complex line drawings, in *Proceedings of 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Vol. 2, pp. 791-796, 2004b.
- Mackay, D.J.C. *Bayesian Methods for Adaptive Models*, PhD. Dissertation, California Institute of Technology, 1991.
- Mackay, D.J.C. A practical Bayesian framework for back-propagation networks, *Neural Computation*, Vol. 4, No. 3, pp. 448-472, 1992.
- Mackay, D.J.C. Bayesian Methods for Backpropagation Networks, *Models of Neural Networks II*, Springer, Domany, E. and Schulten, K. (Eds.), 1993.
- Mackay, D.J.C. Probable networks and plausible predictions-a review of practical Bayesian methods for supervised neural networks, *Network: Computation in Neural Systems*, Vol. 6, No. 3, pp. 469-505, 1995.
- Malik, J. Interpreting line drawing of curved objects, *International Journal of Computer Vision*, Vol. 1, No. 1, pp. 73-103, 1987.
- Mäntylä, M. *An Introduction to Solid Modeling*, Computer Science Press, Rockville, Maryland, US, 1988.
- Marill, T. Emulating the human interpretation of line drawings as three dimensional objects, *International Journal of Computer Vision*, Vol. 6, No. 2, pp. 147-161, 1991.
- Markowsky, G. and Wesley, M.A. Fleshing out wire-frames, *IBM Journal of Research and Development*, Vol. 24, No. 5, pp. 582-597, 1980.
- Marti, E., Regomcós, J., López-Krahe, J. and Villanueva, J. J. Hand line drawing interpretation as three-dimensional object, *Signal Processing*, Vol. 32, No. 1-2, pp. 91-110, 1993.
- McCalley, J.D., Zhou, G.Z. and Acker, V.V. Power system security boundary visualization using neural networks, *Neurocomputing*, Vol. 23, No. 1-3, pp. 85-96, 1998.
- Mckim, R. H. *Experiences in Visual Thinking*(2nd ed.), Brooks/Cole, Monterey, 1980.

References

- Mercer, J. Functions of positive and negative type and their connection with the theory of integral equations, *Philos. Trans. Roy. Soc. London A*, Vol. 209, pp.415-446, 1909.
- Meyer, R.R. and Heindl, E. Reconstruction of off-axis electron holograms using a neural net, *Journal of Microscopy*, Vol. 191, No. 1, pp. 52-59, 1998.
- Milidiú, R.L., Machado R.J. and R.P. Rentería. Time-series forecasting through wavelets transformation and a mixture of expert models, *Neurocomputing*, Vol. 28, No. 1-3, pp. 145-156. 1999.
- Montgomery, D.C. and Peck, E.A. *Introduction to Linear Regression Analysis*, John Wiley & Sons, New York, 1982.
- Neal, R.M. *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics, Vol. 118, Springer, New York, 1996.
- Nezis, K. and Vosniakos, G. Recognizing 2D/3D shape features using a neural network and heuristics, *Computer Aided Design*, Vol. 29, No. 7, pp. 523-539, 1997
- Oh, B. and Kim, C. Progressive reconstruction of 3D objects from a single free-hand line drawing, *Computers and Graphics*, Vol. 27, No. 4, pp. 581-592, 2003.
- Oliver, B., Miguel, E.L. and Andre, H.S. A multi-layered architecture for sketch-based interaction within virtual environments, *Computers and Graphics*, Vol. 24, No. 6, pp. 851-867, 2000.
- Osuna, E., Freund, R., and Girosi, F. Training support vector machines: an application to face detection, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 130-136, 1997.
- Owen C.L. An algorithm for the decomposition of non-directed Graphs, *Emerging methods in Environmental Design and Planning*, Moore, G.T. (Ed.), MIT Press, Cambridge, Mass, pp. 133-146, 1970.
- Pal, N.R. and Pal, S.K. A review on image segmentation techniques, *Pattern Recognition*, Vol. 26, No. 9, pp. 1277-1294, 1993.
- Palmer, S.E. *Vision Science*, MIT Press, Cambridge, MA, 1999.
- Parodi, P., Lancewicki, R., Vijn, A. and Tsotsos, J.K. Empirically-derived estimates of the complexity of labeling line drawings of polyhedral scenes, *Artificial Intelligence*, Vol. 105, No. 1-2, pp. 47-75, 1998.
- Peng, J. and Heisterkamp, D.R. Kernel indexing for relevance feedback image retrieval, in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 733-736, 2003.
- Penny, W. and Roberts, S. J. Bayesian classification using neural networks - How useful is the evidence framework, *Neural Networks*, Vol. 12, No. 6, pp. 877-892, 1999.
- Pereira, J., Jorge, J., Branco, V. and Nunes, F. Towards calligraphic interfaces: sketching 3D scenes with gestures and context icons, in *Proceedings of the 8th*

References

- International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*, (WSCG), Plzen, Czech Republic, 2000.
- Peters, T. J. *Encoding Mechanical Design Features for Recognition via Neural Nets*, Technical Report, No.CSE-TR-91-20, Department of computer science, University of Connecticut, 1991.
- Piperakis, E. and Kumazawa, I. Affine transformations of 3D objects represented with neural networks, in *Proceedings 3-D Digital Imaging and Modeling*, pp. 213-223, 2001.
- Piquer, A., Martin, R.R., and Company, P. Using skewed mirror symmetry for optimisation-based 3D line-drawing recognition, in *Proceedings of IAPR International Workshop on Graphics Recognition (GREC'2003)*, pp.182-193, 2003.
- Pontil, M. and Verri, A. Support vector machines for 3D object recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 6, pp. 637-646, 1998.
- Posch, S. Detecting skewed symmetries, in *Proceeding of the International Conference on Pattern Recognition*, pp. 602-606, 1992.
- Qin, S.F., Wright, D.K. and Jordanov, I.N. From on-line sketching to 2D and 3D geometry: a system based on fuzzy knowledge, *Computer Aided Design*, Vol. 32, No. 14, pp. 851-866, 2000.
- Ragg, T. and Gutjahr, S. Automatic determination of optimal network topologies based on information theory and evolution, in *Proceedings of the 23rd EUROMICRO Conference '97 New Frontiers of Information Technology*, pp. 549-555, 1997.
- Reeves, C. Training set selection in neural network applications, in *Proceedings of the 2nd International Conference on Artificial Neural Nets and Genetic Algorithms*, Springer-Verlag, Vienna, pp.476-478, 1995
- Requicha, A.A.G. *Mathematical Models of Rigid Solids*, Technical Memo No. 28, Production Automation Project, University of Rochester, 1977.
- Riesenhuber, M. and Poggio, T. *A Note on Object Class Representation and Categorical Perception*, Technical Report AI Memo 1679, CBCL Paper 183, MIT AI Lab and CBCL, Cambridge, MA, 1999.
- Rodgers, P.A., Green, G. and McGown, A. Using concept sketches to track design progress, *Design Studies*, Vol. 21, No. 5, pp. 451-464, 2000.
- Samuel, M.M., Requicha, A.A.G. and Elkind, S.A. *Methodology and Results of an Industrial Parts Survey*, Technical Memorandum 21, Production Automation Project, University of Rochester, NY, US, 1976.
- Sankar, P.V. A vertex coding scheme for interpreting ambiguous trihedral solids, *Computer Graphics and Image Processing*, Vol. 6, pp. 61-89, 1977.
- Schoenberg, I. J. Contributions to the problem of approximation of equidistant data by analytic functions, Part A: On the problem of smoothing or graduation, a first class of

References

analytic approximation formulae, *Quarterly of Applied Mathematics*, Vol. 4, No 1, pp. 45-99, 1946a.

Schoenberg, I. J. Contributions to the problem of approximation of equidistant data by analytic functions, Part B: On the problem of osculatory interpolation, a second class of analytic approximation formulae, *Quarterly of Applied Mathematics*, Vol. 4, No 2, pp. 112-141, 1946b.

Schon, D.A. *The Reflective Practitioner: How Professions Think in Action*, Basic Books, New York, 1983.

Schweikardt, E. and Mark D.G. Digital clay: deriving digital models from freehand sketches, *Automation in Construction*, Vol. 9, No. 2, pp. 107-115, 2000.

Sen, A. and Serivastava, M. *Regression Analysis: Theory, Methods, and Applications*, Springer-Verlag, New York, 1990.

Shen, D.G., Ip. H.H.S. and Teoh, E.K. Robust detection of skewed symmetries by combining local and semi-local affine invariants, *Pattern Recognition*, Vol. 34, No. 7, pp. 1417-1428, 2001.

Shevade, S.K., Keerthi, S.S., Bhattacharyya, C. and Murthy, K.R.K. Improvements to SMO algorithm for SVM regression, *IEEE Transactions on Neural Networks*, Vol. 11, No. 5, pp. 1188-1193, 2000.

Shpitalni, M. and Lipson, H. Identification of faces in a 2D line drawing projection of a wire frame object, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 10, pp. 1000-1012, 1996.

Siedlecki, W. and Sklansky, J. On Automatic Feature Selection, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 2, No. 2, pp. 197-220, 1988.

Sittas, E. 3D design reference framework, *Computer Aided Design*, Vol. 23, No. 5, pp. 380-384, 1991.

Smola, A. J. *Regression Estimation with Support Vector Learning Machines*, Master's thesis, Technische Universität München, 1996.

Sollich, P. and Saad, D. Learning from queries for maximum information gain in imperfectly learnable problems, *Advances in Neural Information Processing System*, Vol. 7, Tesauro, G., Touretzky, D.S. and Leen, T.K. (Eds.), pp. 287-294, MIT Press, 1995.

Srinivasan, V., Han, Y.K. and Ong, S.H. Image reconstruction by a Hopfield neural network, *Image and Vision Computing*, Vol. 11, No. 5, pp. 278-282, 1993.

Steppe, J.M. and Bauer, Jr.K.W. Improved feature screening in feed-forward neural networks, *Neurocomputing*, Vol. 13, No. 1, pp. 47-58, 1996.

Stig, O. Qualified product concept design needs a proper combination of pencil-aided design and model-aided design before product data management, *Journal of Engineering Design*, Vol. 9, No. 2, pp. 107-119, 1998.

References

- Sugihara, K. Picture language for skeletal polyhedra, *Computer Graphics and Image Processing*, Vol. 8, pp. 382-405, 1978.
- Sugihara, K. Mathematical structures of line drawings of polyhedrons – towards man-machine communication by means of line drawings, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 4, pp. 458-469, 1982.
- Sugihara, K. An algebraic approach to shape-from-image problems, *Artificial Intelligence*, Vol. 23, No. 1, pp. 59-95, 1984.
- Sugihara, K. *Machine Interpretation of Line Drawings*, MIT Press, Cambridge, 1986.
- Sutherland, I.E. *Sketchpad: A Man-machine Graphical Communication System*, PhD. Dissertation, MIT, 1963.
- Tao, D., Liu J. and Tang, X. Learning user's perception using region-based SVM for content-based image retrieval, in *Proceedings of the International Conference on Imaging Science, Systems and Technology (CISST '04)*, Arabnia, H. R. (Ed.), Las Vegas, Nevada, US, CSREA Press, pp. 462-468, 2004.
- Tay, F.E.H. and Cao, L.J. Improved financial time series forecasting by combining support vector machines with self-organizing feature map, *Intelligent Data Analysis*, Vol. 5, No. 4, pp.1-16, 2001.
- Thomas, G.D. Approximate statistical test for comparing supervised classification learning algorithms, *Neural Computation*, Vol. 10, No. 7, pp. 1895-1923, 1998.
- Tong, S. and Chang, E., Support vector machine active learning for image retrieval, in *Proceedings of the ninth ACM international conference on Multimedia table of contents*, Vol. 9, Ottawa, Canada, pp. 107-118, 2001.
- Trentin, E. and Gori, M. A survey of hybrid ANN/HMM models for automatic speech recognition, *Neurocomputing*, Vol. 37, No. 1-4, pp. 91-126, 2001.
- Turner, A., Chapman, D. and Penn, A. Calligraphic interfaces: sketching space, *Computers and Graphics*, Vol. 24, No.12, pp. 869-879, 2000.
- Turner, K. *Computer Perception of Curved Objects using a Television Camera*, PhD. Dissertation, Edinburgh University, 1974.
- Ullman, D.G., Wood, S. and Craig, D. The Importance of drawing in the mechanical design process, *Computers and Graphics*, Vol. 14, No. 2, pp. 263-274, 1990.
- Ullman, S. *High-level Vision: Object Recognition and Visual Cognition*, MIT Press, Cambridge, MA, 1996.
- Vafaie, H. and Jong, D.K. Robust feature selection algorithms, in *Proceedings of the Fifth International Conference on Tools with Artificial Intelligence (ICTAI '93)*, US, pp. 356-363, 1993.
- Van, G.T., Suykens, J.A.K., De Moor, B. and Vandewalle, J. Automatic relevance determination for least squares support vector machine classifiers, in *Proceedings of the 9th European Symposium on Artificial Neural Networks*, pp. 13-18, 2001.
- Vapnik, V.N. *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

References

- Vapnik, V.N. *The Nature of Statistical Learning Theory* (2nd ed.), Springer, New York, 2000.
- Vapnik, V.N. and Chapelle, O. Bounds on error expectation for support vector machine, *Neural Computation*, Vol. 12, No. 9, pp. 2013-2036, 2000.
- Varley, P.A.C. and Martin R.R. The junction catalogue for labelling line drawings of polyhedra with tetrahedral vertices”, *International Journal of Shape Modeling*, Vol. 7, No. 1, pp. 23-44, 2001.
- Varley, P.A.C. and Martin, R.R. A system for constructing boundary representation solid models from a two-dimensional sketch — frontal geometry and sketch categorization, in *Proceedings of 1st UK-Korea Workshop on Geometric Modeling and Computer Graphics*, Kyung Moon Publishers, pp. 113-128, 2000a.
- Varley, P.A.C. and Martin, R.R. A system for constructing boundary representation solid models from a two-dimensional sketch — topology of hidden parts, in *Proceedings of 1st UK- Korea Joint Workshop on Geometric Modeling and Computer Graphics*, Kyung Moon Publishers, pp. 129-144, 2000b.
- Varley, P.A.C. and Martin, R.R. A system for constructing boundary representation solid models from a two-dimensional sketch — geometric finishing, in *Proceedings of 1st UK-Korea Workshop on Geometric Modeling and Computer Graphics*, Kyung Moon Publishers, pp. 145-158, 2000c.
- Varley, P.A.C. and Martin, R.R. Estimating depth from line drawings, in *Proceedings of 7th ACM Symposium on Solid Modeling and Applications* (SM02), ACM Press, pp. 180-191, 2002.
- Varley, P.A.C. *Automatic Creation of Boundary-Representation Models from Single Line Drawings*, PhD. Dissertation, Department of Computer Science, College of Cardiff, University of Wales, 2002.
- Walderon, M.B. and Walderon, K.J. Conceptual CAD tools for mechanical engineers, in *Proceedings of Computers in Engineering Conference*, Patton E.M. (Ed.), *Computer and Graphics*, Vol. 2, pp. 203-209, 1988.
- Waltz, D.L. *Generating Semantic Descriptions from Drawings of Scenes with Shadows*, Technical Report AITR 271, MIT, Cambridge, MA, 1972.
- Wang, Y., Chua, C. and Ho, Y. Facial feature detection and face recognition from 2D and 3D images, *Pattern Recognition Letters*, Vol. 23, No. 10, pp. 1191-1202, 2002.
- Wang, Y.M. and Wahl, F.M. Vector-entropy optimization-based neural-network approach to image reconstruction from projections, *IEEE Transactions on Neural Networks*, Vol. 8, No. 5, pp. 1008-1014, 1997.
- Wei, G.Q. and Hirzinger, G. Learning shape from shading by a multilayer network, *IEEE Transactions on Neural Networks*, Vol. 7, No. 4, pp. 985-995, 1996.
- Weiler, K. *Topological Structures for Geometrical Modeling*, PhD. Dissertation, Rensselaer Polytechnic Institute, 1986.

References

- Wermter, S. *Hybrid Approaches to Neural Network-based Language Processing*, Technical Report TR-97-030, International Computer Science Institute, Berkeley, California, 1997.
- Weston, J. and Watkins, C. *Multi-class Support Vector Machines*, Technical Report CSD-TR-98-04, Royal Holloway, University of London, UK, 1998.
- Whitney W. Motions and stresses of projected polyhedra, *Structural Topology*, Vol. 7, pp. 13-38, 1982.
- Williams, C.K.I. and Barber, D. Bayesian classification with Gaussian processes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, pp. 1342-1351, 1998.
- Wu, T. *Developing Blackboard System for Product Design Using a Graph Decomposition Algorithm*, Master's thesis, Nanyang Technological University, 2001.
- Yamany, S.M., Khiani K.J. and Farag, A. Application of neural networks and genetic algorithms in the classification of endothelial cells, *Pattern Recognition Letters*, Vol. 18, No. 11-13, pp. 1205-1210. 1997.
- Yang, M. and Lee, E. Improved neural network model for reverse engineering, *International Journal of Production Research*, Vol. 38, No. 9, pp. 2067-2078, 2000.
- Yann, F. and Bahram J. Neural network for three-dimensional object recognition based on digital holography, *Optics Letters*, Vol. 26, No. 19, pp. 1478-1480, 2001.
- Yip, R.K.K., Tam, P.K.S. and Leung, D.N.K. Application of elliptic Fourier descriptors to symmetry detection under parallel projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 3, pp 277-286, 1994.
- Zakarian, V.L. and Kaiser, M.J. An embedded hybrid neural network and expert system in a computer-aided design system, *Expert Systems with Applications*, Vol. 16, No. 2, pp. 233-243, 1999.
- Zelevnik, R.C., Herndon, K.P. and Hughes, J.F. SKETCH: An interface for sketching 3D scenes, in *SIGGRAPH'96 Conference Proceedings*, pp. 163-170, 1996.
- Zhang, L., Lin, F.Z. and Zhang, B. Support vector machine learning for image retrieval, in *Proceedings of IEEE International Conference on Image Processing (ICIP 2001)*, Vol. 2, pp. 721-724, 2001.
- Zhao, W., Chellappa, R., Phillips, P.J. and Rosenfeld, A., Face recognition: A literature survey, *ACM Computing Surveys (CSUR)*, Vol. 35, No. 4, pp. 399-458, 2003.
- Zurada, J., Malinowski, M.A. and Usui, S. Perturbation method for deleting redundant inputs of perceptron networks, *Neurocomputing*, Vol. 14, No. 2, pp. 177-193, 1997.

APPENDIX A

Support Vector Regression Estimation

A.1 Definition of regression

Suppose we are given a set of observations generated from an unknown probability distribution $P(\mathbf{x}, y)$

$$X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l)\} \text{ with } \mathbf{x}_i \in \mathbf{R}^n, y_i \in \mathbf{R} \quad (\text{A.1})$$

and a class of functions $F = \{f | f : \mathbf{R}^n \mapsto \mathbf{R}\}$

then the basic regression problem is to find a function $f \in F$ that minimizes a risk function

$$R(f) = \int l(y - f(\mathbf{x}), \mathbf{x}) dP(\mathbf{x}, y) \quad (\text{A.2})$$

where l is a loss function, it indicates how differences between y and $f(x)$ should be penalized.

The ultimate goal is to use the estimator $\hat{f}$ to accurately predict future values for new unseen data patterns. In regression estimation, the unseen data patterns are called test samples. The sum of deviations between the actual values and estimated values on the test samples is called the test error. The sum of deviations between the actual values and estimated values on the training samples is called the training error. For this goal, the estimator should have not only a small training error, but also a small test error. If the estimator has a small training error but a large test error, it is called over-fitting. If the estimator over-fits the training samples, it will have poor

generalization performance — the ability of the estimator to predict future values to unseen data patterns is poor.

A.2 KKT conditions in SVR problem

A reasonable understanding of nonlinear optimization principle about Karush-Kuhn-Tucker (KKT) conditions is needed for appreciating some important steps of the SVR computation. A suitable way is needed to solve Equation 3.3, which is formulated as an optimization problem for the regression risk. KKT conditions are used to find the solution. For convex problems with linear constraints, i.e., the problem defined in Equation 3.3, KKT conditions are both necessary and sufficient because a point is optimal if and only if the KKT conditions hold at that point.

The KKT theorem of SVR problem (Kuhn and Tucker, 1951), at the saddle point is listed as follows:

$$\alpha_i (\varepsilon + \xi_i - y_i + \mathbf{w} \cdot \varphi(\mathbf{x}_i) + b) = 0 \quad (\text{A.3})$$

$$\alpha_i^* (\varepsilon + \xi_i^* + y_i - \mathbf{w} \cdot \varphi(\mathbf{x}_i) - b) = 0 \quad (\text{A.4})$$

$$(C - a_i^{(*)}) \xi_i^{(*)} = 0 \quad (\text{A.5})$$

$$\alpha_i \alpha_i^* = 0, \xi_i \xi_i^* = 0, i = 1, \dots, l. \quad (\text{A.6})$$

From (A.3) - (A.6), three conditions can be drawn:

- (1) According to equation (A.5), only one of the variables $a_i^{(*)}$ can be non-zero for any training data point.
- (2) According to (A.3), $\xi_i^{(*)} \neq 0$ only when $a_i^{(*)} = C$. This means that only the training data points with $a_i^{(*)} = C$ could lie outside the ε -insensitive tube.

Appendix A Support Vector Regression Estimation

(3) b can be calculated from the training data points with $0 < a_i^{(*)} < C$. According to

(A.5), $\xi_i^{(*)} = 0$ for $0 < a_i^{(*)} < C$. From (A.3) and (A.4), b can be calculated as follows:

$$b = y_i - \mathbf{w} \cdot \phi(\mathbf{x}_i) - \varepsilon \quad \text{if} \quad 0 < a_i < C \quad (\text{A.7})$$

$$b = y_i - \mathbf{w} \cdot \phi(\mathbf{x}_i) + \varepsilon \quad \text{if} \quad 0 < a_i^* < C \quad (\text{A.8})$$

Furthermore, to satisfy the KKT optimal conditions (A.3) and (A.4), both a_i and a_i^* will assume zero for the training data points with $|y_i - \mathbf{w} \cdot \phi(\mathbf{x}_i) - b| < \varepsilon$ as the second terms of (A.3) and (A.4) are non-zero: $(\varepsilon + \xi_i - y_i + \mathbf{w} \cdot \phi(\mathbf{x}_i) + b) \neq 0$ and $(\varepsilon + \xi_i^* + y_i - \mathbf{w} \cdot \phi(\mathbf{x}_i) - b) \neq 0$. In other words, only for the training data points with $|y_i - \mathbf{w} \cdot \phi(\mathbf{x}_i) - b| \geq \varepsilon$, a_i or a_i^* may assume non-zero values. The training data points with $(a_i - a_i^*) \neq 0$ are called support vectors. From (A.5), it can be concluded that only the support vectors are used to determine the decision function as the values of $(a_i - a_i^*)$ for the other training data points are equal to zero. As support vectors are usually only a small subset of the training data points, this characteristic of SVR is referred to as the sparsity of the solution. This sparsity of the solution provides fast convergence speed in the training of SVR, due to the use of the ε -insensitive loss function.

A.3 Training algorithm

SVR training algorithm is called sequential minimal optimization (SMO) which has been widely used due to fast learning speed even in large problems. Since it always optimizes and alters two Lagrange multipliers at every step, making the SVR

Appendix A Support Vector Regression Estimation

optimization problem solvable easily and quickly. The improved SMO learning algorithm is introduced as following (Shevade et al., 2000).

Define the following index sets at a given $a_i^{(*)}$:

$$I_{0a} = \{i : 0 < a_i < C\}; I_{0b} = \{i : 0 < a_i^* < C\}; I_1 = \{i : a_i = 0, a_i^* = 0\};$$

$$I_2 = \{i : a_i = 0, a_i^* = C\}; I_3 = \{i : a_i = C, a_i^* = 0\}; I_0 = I_{0a} \cup I_{0b}.$$

$\tilde{F}_i$ and $\bar{F}_i$ are defined as

$$\tilde{F}_i = F_i + \varepsilon \quad \text{if } i \in I_{0b} \cup I_2, \quad \tilde{F}_i = F_i - \varepsilon \quad \text{if } i \in I_{0a} \cup I_1; \quad (\text{A.9})$$

$$\bar{F}_i = F_i - \varepsilon \quad \text{if } i \in I_{0a} \cup I_3, \quad \bar{F}_i = F_i + \varepsilon \quad \text{if } i \in I_{0b} \cup I_1. \quad (\text{A.10})$$

where $F_i = y_i - \sum_{j=1}^N (a_j - a_j^*) K(\mathbf{x}_i, \mathbf{x}_j)$, and it is available for all i at any instant.

Let i_{low} and i_{up} be indices such that

$$\tilde{F}_{i_{low}} = b_{low} = \max\{\tilde{F}_i : i \in I_0 \cup I_1 \cup I_2\}; \quad (\text{A.11})$$

$$\bar{F}_{i_{up}} = b_{up} = \min\{\bar{F}_i : i \in I_0 \cup I_1 \cup I_3\}. \quad (\text{A.12})$$

Then optimality conditions will hold at $a^{(*)}$ iff

$$b_{low} \leq b_{up} + 2\tau \quad (\text{A.13})$$

where τ is a positive tolerance parameter. Correspondingly, violation at a and a^* is defined if one of the following two conditions holds:

$$i \in I_0 \cup I_1 \cup I_2, \quad j \in I_0 \cup I_1 \cup I_3 \quad \text{and} \quad \tilde{F}_i > \bar{F}_j + 2\tau \quad (\text{A.14})$$

$$i \in I_0 \cup I_1 \cup I_3, \quad j \in I_0 \cup I_1 \cup I_2 \quad \text{and} \quad \bar{F}_i < \tilde{F}_j - 2\tau \quad (\text{A.15})$$

The basic step in the SMO algorithm is to choose a pair of indexes (i_1, i_2) and optimize the objective function by varying the Lagrange multipliers corresponding to i_1 and i_2 . After a successful step using a pair of indices (i_1, i_2) , let $\tilde{I} = I_0 \cup \{i_1, i_2\}$ and

Appendix A Support Vector Regression Estimation

use it to compute (i_{low}, b_{low}) and (i_{up}, b_{up}) . This algorithm always begins with the worst violating pair, i.e. choosing $i_2 = i_{low}$ and $i_1 = i_{up}$. For a given i , F_i is computed first and optimality is checked using the current (i_{low}, b_{low}) and (i_{up}, b_{up}) . If there is no violation, F_i is used to update these quantities. Otherwise, (i, i_{low}) or (i_{up}, i) is used in terms of the violation case.

This algorithm first loops over the entire training set once, then makes repeated passes over the samples with $(0 < a_i < C$ or $0 < a_i^* < C)$ until all of those samples obey the optimality conditions. Again the algorithm alternates between single pass over the entire training set and multiple passes over the non-bound subset until the number of a_i and a_i^* needed to be changed in the entire training set is equal to zero.

APPENDIX B

Object Library

B. 1 Class 1

Appendix B Object Library

Appendix B Object Library

B.2 Class 2

Appendix B Object Library

B.3 Class 3

Appendix B Object Library

B.4 Class 4

APPENDIX C

Reconstruction Results from Optimal Regularity Set

Model No	Input line drawing	Reconstruction result	
11
12
13
14
15

Appendix C Reconstruction Results from Optimal Regularity Set

16
17
18
19
20
21
22
23

APPENDIX D

T-test for 3D Reconstruction

A T-test (Koopmans, 1987) is used to determine whether the two "independent" groups significantly differ. In this thesis, it is used to compare the reconstruction error results between the two sample sets. T-test is used separately in Chapter 4 and 5 for evaluation.

D.1 T-test for reduced regularity set

A T-test is used to compare the performance of the selected and the original regularity sets S1 and S2 in the 3D reconstruction. The two sets are defined in Section 4.5.1. 23 samples in Table 4.3 are selected to do the test. E_1 and E_2 are the reconstruction errors of the regularity set S1 and S2 separately.

1. Null hypothesis: $E_2 - E_1 = 0$

Alternative hypothesis: $E_2 - E_1 > 0$

2. Level of significance $\alpha = 0.1$
3. Criterion: Reject the null hypothesis if $Z > 1.3213$, where 1.3213 is the value of $t_{0.1}$ for 22 degrees of freedom and where Z is given by the samples formula below:

$$Z = \frac{(\bar{E}_2 - \bar{E}_1)}{\sqrt{\frac{S_2^2}{n} + \frac{S_1^2}{n}}}$$

Appendix D T-test for 3D Reconstruction

where $\bar{E}$ denotes the average reconstruction error based on the samples, S denotes the standard deviation of reconstruction error and n is the degree of freedom.

4. Calculation:

$$Z = \frac{0.0550 - 0.0544}{\sqrt{\frac{0.0490^2}{22} + \frac{0.0489^2}{22}}} = 0.0407$$

5. Decision: Since $Z = 0.0407$ is smaller than 1.3213, the null hypothesis cannot be rejected. Thus, the reconstruction error from the selected regularity set S2 is not larger than of the original set S1. The selected regularity set S2 is accepted as an optimal selection for the reconstruction.

D.2 T-test for hybrid intelligent system

The single reconstruction system with the regularity set {R1, R2, R3, R4, R9} is used as the benchmark. It is used to compare the reconstruction performance with the proposed hybrid system (Section 5.7) on the normalized standard deviation error.

The benchmark is the single reconstruction system applying a reduced regularity set {R1, R2, R3, R4, R9}, and the functional components in the hybrid system refer to the reconstruction experts 1 and 2. Nine samples in the Classes 1 and 2 (Objects 1-9 listed in Table 5.6) are used to do the test. E_1 and E_2 are the reconstruction error obtained from the benchmark and the hybrid system separately.

1. Null hypothesis: $E_2 - E_1 = 0$

Alternative hypothesis: $E_2 - E_1 > 0$

2. Level of significance $\alpha = 0.1$.

3. Criterion: Reject the null hypothesis if $Z > 1.86$, where 1.86 is the value of $t_{0.1}$ for n degrees of freedom and where Z is given by the samples formula below:

Appendix D T-test for 3D Reconstruction

$$Z = \frac{(\bar{E}_2 - \bar{E}_1)}{\sqrt{\frac{S_2^2}{n} + \frac{S_1^2}{n}}}$$

where $\bar{E}$ denotes the average reconstruction error based on the selected samples in Table 5.6, S denotes the standard deviation of reconstruction error and n is the degree of freedom, $n=9-1$.

4. Calculation:

$$Z = \frac{0.081 - 0.086}{\sqrt{\frac{0.046^2}{8} + \frac{0.045^2}{8}}} = -0.22$$

5. Decision: Since $Z = -0.22$ is smaller than 1.86, the null hypothesis cannot be rejected. Thus, the reconstruction error from the reconstruction expert 1 and 2 is no greater than that of the benchmark system.

APPENDIX E

SVM Theory in Classification Task

E.1 Basic theory of SVM

In the past few years, there has been an increasing development in SVM – an optimization method to predict the output of unseen datasets. In SVM, the support vectors come from elements of a dataset which are specifically chosen for the classification task, and considered important in separating two classes from each other. Meanwhile, other input vectors are ignored because of their insignificance in constructing the optimal hyperplane. With the use of the structural risk minimization principle (Haykin, 1999), SVM gives good generalization performance on practical problems.

SVM was initially developed by Vapnik (1995) for the classification problem. As a supervised method, SVM can take advantage of prior knowledge of drawing recognition and construct a hyperplane as the decision surface so that the margin of the separation between different drawing samples is maximized.

To explain the basic idea behind SVM, we start with the simplest case: linear machine trained on separable data.

E.1.1 Linear SVM

In the classification problem, a hyperplane is a linear function that is capable of separating the training data without error. The minimal distance from the hyperplane to the closest data point is called the margin and the separating hyperplane with the

Appendix E SVM Theory in Classification Task

maximum margin is called optimal hyperplane. It is intuitively clear that a larger margin corresponds to a better generalization.

Let's consider a binary classification task with l training data $\mathbf{x}_i \in R^d$ ($i = 1, \dots, l$) having corresponding class labels $y_i = \pm 1$. Suppose that the training data can be separated by a hyperplane decision function (E.1) with appropriate coefficients $\mathbf{w}$ and b , with $\mathbf{x}$ as an input vector.

$$g(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b \quad (\text{E.1})$$

The problem of finding an optimal hyperplane is to find $\mathbf{w}$ and b that maximize the margin. Using the method of Lagrange multipliers, the quadratic optimization problem with linear constraints may be formally stated below.

Given the training samples $(\mathbf{x}_i, y_i), i = 1, \dots, l$, find the parameter α_i to maximize the objective function

$$L_D(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad (\text{E.2})$$

$$\text{subject to constraints} \quad \sum_{i=1}^l y_i \alpha_i = 0 \quad (\text{E.3})$$

$$\alpha_i \geq 0, \quad i = 1, \dots, l \quad (\text{E.4})$$

This leads to a hyperplane in the form

$$g(\mathbf{x}) = \sum_{i=1}^l \alpha_i^0 y_i (\mathbf{x} \cdot \mathbf{x}_i) - b_0 \quad (\text{E.5})$$

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad (\text{E.6})$$

where α_i^0 are Lagrange multipliers at the solution of the optimization problem (E.2)-(E.4) and the parameter b_0 at the solution is computed by taking advantage of the conditions on the support vectors.

E.1.2 Nonlinear SVM

As the computation capability of the linear functions is somewhat limited, extending the linear SVM into nonlinear is more practical for solving complex problems.

The idea of generating nonlinear SVM is to map the original input space X into a high dimensional feature space H by a function $\phi(\mathbf{x})$ and then to construct a linear function in the high dimensional feature space which corresponds to a nonlinear function in the original input space.

Both the decision function $g(\mathbf{x})$ and optimization problem remain in the same form except that the input vector $\mathbf{x}$ is replaced by $\phi(\mathbf{x})$, or in other words, the inner product $(\mathbf{x}_i \cdot \mathbf{x}_j)$ is replaced by $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$.

With the increasing dimension of the used feature space, computation cost will increase. To efficiently compute the linear function which is defined in the high dimensional feature space, the kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ is used, where $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$. The inner products $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ are all replaced with the kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. The mapping of $\phi(\mathbf{x})$ from the original input space X is implicit, which is not proportional to the dimension of $\phi(\mathbf{x})$. The use of the kernel functions makes it possible to map the data implicitly into a high dimensional feature space.

Any function satisfying Mercer's condition (Mercer, 1909) can be used as the kernel function. The commonly used kernel function is Gaussian kernel.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{\delta^2}(\mathbf{x}_i - \mathbf{x}_j)^2\right) \quad (\text{E.7})$$

Appendix E SVM Theory in Classification Task

By using the kernel functions, SVM can efficiently and effectively construct many types of nonlinear functions. The detailed nonlinear SVM classification with soft boundary is introduced in Section 5.6.1.

E.2 Network structure and training method

Figure E.1 shows the network structure of SVM, which includes input layer, output layer and hidden layer (Haykin, 1999).

Figure E.1 SVM Configuration.

The SVM training algorithm adopted is a sequential minimal optimization (SMO) which has been widely used due to its fast learning speed even for large problem. Since it always optimizes and alters two Lagrange multipliers at every step, the problem can be solved more easily and quickly. The improved SMO learning algorithm is introduced as follows (Shevade et al., 2000).

Define the following index sets at a given α :

$$I_0 = \{i : 0 < \alpha_i < C\}; \quad I_1 = \{i : y_i = 1, \alpha_i = 0\}; \quad I_2 = \{i : y_i = -1, \alpha_i = C\};$$

$$I_3 = \{i : y_i = 1, \alpha_i = C\}; \quad I_4 = \{i : y_i = -1, \alpha_i = 0\}. \quad (\text{E.8})$$

Define $F_i = \sum_{j=1}^N \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) - y_i$, and

Appendix E SVM Theory in Classification Task

$$F_{i_up} = b_{up} = \min\{F_i : i \in I_0 \cup I_1 \cup I_2\};$$

$$F_{i_low} = b_{low} = \max\{F_i : i \in I_0 \cup I_3 \cup I_4\}. \quad (E.9)$$

Then optimality conditions will hold at some α iff

$$b_{low} \leq b_{up} + 2\tau \quad (E.10)$$

where τ is a positive tolerance parameter.

Correspondingly, violation is defined at α if one of the following sets of conditions holds:

$$i \in I_0 \cup I_3 \cup I_4, \quad j \in I_0 \cup I_1 \cup I_2 \text{ and } F_i > F_j + 2\tau \quad (E.11)$$

$$i \in I_0 \cup I_1 \cup I_2, \quad j \in I_0 \cup I_3 \cup I_4 \text{ and } F_i < F_j - 2\tau \quad (E.12)$$

This algorithm always begins with the worst violating pair. Choose $i_2=i_{low}$ and $i_1=i_{up}$.

If the target y_1 does not equal the target y_2 , then the following bounds apply to α_2 :

$$L=\max(0, \alpha_2-\alpha_1), \quad H=\min(C, C+\alpha_2-\alpha_1) \quad (E.13)$$

If the target y_1 equals to the target y_2 , then the following bounds apply to α_2 :

$$L=\max(0, \alpha_2+\alpha_1-C), \quad H=\min(C, \alpha_2+\alpha_1) \quad (E.14)$$

The second derivative of the objective function along the diagonal line can be expressed as:

$$\eta = \mathbf{K}(\mathbf{x}_1, \mathbf{x}_1) + \mathbf{K}(\mathbf{x}_2, \mathbf{x}_2) - 2\mathbf{K}(\mathbf{x}_1, \mathbf{x}_2) \quad (E.15)$$

If η is positive, $\alpha_2^{new} = \alpha_2 + y_2(F_1 - F_2)/\eta$, then

$$\alpha_2^{new, clipped} = \begin{cases} H & , \quad \alpha_2^{new} \geq H; \\ \alpha_2^{new} & , \quad L < \alpha_2^{new} < H; \\ L & , \quad \alpha_2^{new} \leq L. \end{cases} \quad (E.16)$$

Otherwise, *SMO* will move the Lagrange multipliers to the end point (L or H) that has the lower value of the objective function.

Appendix E SVM Theory in Classification Task

And if $|\alpha_2^{new,clipped} - \alpha_2| \geq eps * (\alpha_2^{new,clipped} + \alpha_2 + eps)$, then

$$\alpha_1^{new} = \alpha_1 + y_1 y_2 * (\alpha_2 - \alpha_2^{new,clipped}) \quad (E.17)$$

where the constant eps is a tolerance parameter. Otherwise, select another different α_2 and repeat the whole process. Note that after a successful step using a pair of indices (i_2, i_1) , let $\tilde{I} = I_0 \cup \{i_1, i_2\}$. (i_{low}, b_{low}) and (i_{up}, b_{up}) are computed using $\tilde{I}$ only. Update F values for i_1 and i_2 by (E.18) and (E.19), and use this together with y_i to decide the index set of i .

$$F_1^{new} = F_1 + y_1 * (\alpha_1^{new} - \alpha_1) * K(\mathbf{x}_1, \mathbf{x}_1) + y_2 * (\alpha_2^{new,clipped} - \alpha_2) * K(\mathbf{x}_1, \mathbf{x}_2) \quad (E.18)$$

$$F_2^{new} = F_2 + y_1 * (\alpha_1^{new} - \alpha_1) * K(\mathbf{x}_1, \mathbf{x}_2) + y_2 * (\alpha_2^{new,clipped} - \alpha_2) * K(\mathbf{x}_2, \mathbf{x}_2) \quad (E.19)$$

This algorithm first loops over the entire training set once, then makes repeated passes over the non-bound examples ($0 < \alpha_i < C$) until all of those examples obey the optimality conditions (E.10). Again the algorithm alternates between a single pass over the entire training set and multiple passes over the non-bound subset until the number of α_i needed to be changed in the entire training set is zero. The convergence of this algorithm has been proved by Keerthi and Gilbert (2002).

E.3 Generalization performance and evaluation criteria

Many practical time series contain a large amount of noise. If the number of free parameters within a network is too large, this network can capture not only useful information contained in the data but also unwanted noises, which means over-fitting. This will certainly lead the network to perform well on the training data, but generalize poorly to similar input-output patterns not used in training. In order to avoid over-fitting, generalization must be evaluated in the learning process.

Appendix E SVM Theory in Classification Task

Generalization performance measures how well a network performs on unseen data after the training stage has been completed. The problem of generalization is analogous to the curve fitting problem with regularization - a good curve fit must represent some underlying trend in the data rather than provide an arbitrarily close fit to the data points without regard to complexity.

Several factors influence the generalization ability of a network: the size and efficiency of the training set, the architecture of the network, the physical complexity of the problem and so on. Of these factors, establishing the training samples that contain sufficient information to enable the network to generalize correctly, and optimizing the structure and architecture of the network have been investigated, the complexity of the problem is the only one over which we have no control. using classification error.

In practical application, the generalization performance is evaluated by three statistical metrics. They are classification error, training time and the number of support vectors. They are explained as follows.

The classification error, in the form of percentage, measures the deviation between the actual and predicted output; a smaller value suggests a better prediction. Classification error can also be used to evaluate the quality of the training set and the network performance.

In SVM, support vectors (SVs) come from the training data which are considered important in separating the two classes from each other, while other input vectors are ignored because of their insignificance in constructing a hyperplane for the classification. Under a similar classification error, a smaller number of SVs is preferred. Consider an example, if a classification task can be constructed by two kinds network structure according to SV sets A and B with similar classification error,

Appendix E SVM Theory in Classification Task

and the size of A is smaller than that of B, then the trained network based on A is considered more effective and powerful to describe this classification task than that of B. Since the decision time closely relates with the number of SV, a shorter decision time is also found with set A.

On the issue of training time, let us consider another example. Under the same SVM network structure, two classification models with a similar classification error are developed based on training data sets C and D from one data population. Assuming the training time of data set C is smaller than that of D, the classification network based on C generalizes more quickly than that of D, therefore training data set C is considered to have a better learning performance.

E.4 Selecting training data and tuning parameters

E.4.1 Training data selection

When NNs and related methods are used in classification, independent parameters in this network will be specified in terms of training data. Hence, the network performance heavily depends upon the quality of training data, assuming an optimal topology for the network is known. In the past, training data is selected arbitrarily. Reeves (1995) found that different training data set gave substantially different error in classification. Sollich and Saad (1995) observed that a smaller training set may produce equal or better generalization performance than a larger data set containing redundant samples. Hence, an effective training data set should be selected, which can not only reduce the size of training data set, but also enhance the computation performance.

Misclassification rate, decision time (the number of support vectors) and training time are key factors to evaluating the quality of training data set in network learning.

E.4.2 Parameters tuning

Choosing optimal hyperparameter values is an important step in SVM design. A hyperparameter is a parameter defined in a hyperplane, such as ξ_i , C etc. This is usually done by minimizing either an estimate of the generalization error or some other related performance measure. In this research, the tradeoff between the lowest misclassification rate and the smallest decision time (the smallest number of support vectors) is the key in parameter selection.

During the past few years, several tuning methods have been proposed. Xi-Alpha Bound developed by Joachims (2000) estimated an upper bound on the error rate of the leave-one-out procedure. Approximate Span Bound introduced by Vapnik and Chapelle (2000) not only can provide a good function for SVM hyperparameter selection, but also reflect the actual error rate. A Vapnik-Chervonenkis (VC) bound was proposed (Burges, 1998) to approximate the VC-dimension in the sum of empirical risk and VC confidence by a loose bound. However, none of them yielded a performance as good as k-fold cross validation (CV), which can provide a good correlation with the test error (Duan et al., 2003a). CV is explained in Section 3.6.2.

APPENDIX F

Pseudo Code for Classification

One-versus-one is adopted for line drawing multi-classification. The one-versus-one approach constructs one binary classifier for every pair of distinct classes, which is discussed in Section 5.6.2. The pseudo code for performing drawing classification is given as follows (Keerthi and Shevade, 2003).

1. Kernel (i, j) refers to $k(x_i, x_j)$.
2. The a_i are stored in alpha array, while the F_i are stored in Fcache array.
3. ε is referred to as epsilon, 10^{-6} is a suitable value for it.
4. eps is used to check if sufficient change has occurred in each step. It can be set to 10^{-12} .
5. The function fabs returns the absolute value of a real number.
6. Alpha[i]: Lagrange multiplier associated with i-th example
7. Any lines in the pseudo code that begin with the character // are comments or explanations of the codes.

procedure takeStep (i1, i2)

alph1 = alpha{i1}; alph2 = alpha{i2}; y1 = y{i1}; y2 = y{i2};

F1 = Fcache{i1}; F2 = Fcache{i2}; gamma = alph1 + alph2;

k11 = kernel(i1,i1); k22 = kernel(i2,i2); k12 = kernel(i1,i2);

eta = 2*k12-k11-k22; a2 = alph2 - (F1-F2)/eta;

if (fabs(a2-alph2) < eps*(a2+alpha2+eps)) return 0;

Appendix F Pseudo Code for Classification

```

    a1 = gamma-a2;      alpha{i1} = a1;      alpha{i2} = a2;

    Update the DualObjectiveFunction.

    Update Fcache{i} for all i in I;

    Compute (i_low, b_low) and (i_up, b_up).

    return 1;

endprocedure

procedure ComputeDualityGap()

    DualityGap = 0;      set b.

    loop over all i;

    xi = b + (alpha{i}/C) - Fcache{i};

    DualityGap += alpha{i}(Fcache{i}-(0.5*alpha{i}/C)) + C*xi*xi/2;

    return DualityGap;

endprocedure

procedure InitializeVariables(delta)

    if delta is non-zero

        // delta denotes the factor by which C is changed.

        for every i, reset Fcache{i}.

        compute DualObjectiveFunction.

    else

        initialize alpha{i}=0 for all i.

        set Fcache{i}=-y{i} for all i and DualObjectiveFunction to 0.

        Compute (i_low, b_low) and (i_up, b_up).

    Endprocedure

```

Appendix F Pseudo Code for Classification

```
procedure SMO()

    //      If solution is available at some value of  $C=C1$  and solution is to be
    //      found at  $C=C2$  then set  $\delta := C2/C1$ . Otherwise, set  $\delta := 0$ .

    InitializeVariables( $\delta$ );

    numChanged = 1;    DualityGap = ComputeDualityGap();

    while (DualityGap > epsilon * DualObjectiveFunction && numChanged != 0)

        numChanged = takeStep( $i\_up$ ,  $i\_low$ );

        DualityGap = ComputeDualityGap();

        //      If the procedure stops with numChanged=0, it means that
        //      epsilon is too small.

endprocedure
```

APPENDIX G

Graph Decomposition

Owen's revised non-directed graph decomposition algorithm is adopted to decompose the graph representing non-manifolds (Chen and Occña, 2000; Chen et al., 2002). The reasons for choosing this algorithm are:

- (1) The algorithm is able to decompose a non-directed graph to several sub-graphs;
- (2) The number of blocks and the number of vertex numbers in each block isn't prescribed;
- (3) The decomposed blocks are joint, which means sub-graph intersections are permitted.

The standard steps (Wu, 2001) for decomposing a graph representing a non-manifold is given as follows.

Step 1: Set the value of the connect ratio, then assign a weight to all the edges (can also be called links) and generate an array to store the weights of link:

$$W_j^l = C_j + \lambda_j \quad (\text{G.1})$$

where W_j^l is edge weight of the j^{th} edge, $j=1, 2, 3, \dots, m$ (m is the total number of edges); λ_j is the weight index for edge j , it depends on the experience values derived from the drawing process; C_j is the number of three-edge circuits in the j^{th} link.

Appendix G Graph Decomposition

Step 2: Assign weights to all the vertices, and generate an array to store these weights:

$$W_i^v = \frac{1}{n_i} \sum_{j=1}^{n_i} W_j^l \quad (\text{G.2})$$

where W_i^v is the weight of the i^{th} vertex, n_i is the number of edges incident to the i^{th} vertex, and W_j^l is the weight of the j^{th} edge.

Step 3: Sort the vertices in descending order by weight, bubble sort is adopted.

Step 4: Apply a strong-link decomposition pass: Construct sub-graph blocks by considering the vertices in order, under the following two conditions: that a vertex may join any block if (1) its connection ratio in the original uncut graph exceeds a threshold value, and (2) it is connected to any of the vertices present in the block by its strongest link (for example the degree of the vertices in the block will be three if this vertex is added). If a vertex does not meet the conditions for entry with any of the blocks already created, permit it to start a new sub-graph block. Ignore disconnected vertices.

Step 5: Apply follow-up decomposition passes: Augment already formed sub-graph blocks by reconsidering all vertices in order (including disconnected vertices), and allowing a vertex to join any block in which it meets condition (1) of

Step 4. Continue passes until no further additions are made in a single pass.

Step 6: Compare the sub-graphs found in the present series of decomposition passes with those saved from previous passes. Save all sub-graphs that are not identical to previously saved sub-graphs.

Step 7: Compute the value of the cutting function, k_i , for a possible i^{th} cutting pass:

$$k_i = \max[k_{i-1} + 1, \min(w_j^l)] \quad (\text{G.3})$$

Appendix G Graph Decomposition

where $j = 1, 2, \dots, n$, with n = the number of weighted edges remaining at the beginning of the i th pass; $i = 1, 2, \dots, n-1$; $k_{i-1} = 0$, if no cuts have yet been made. Compare k_i with $\max(w_j^l)$. If $k_i \geq \max(w_j^l)$, go to Step 8; otherwise, apply a cutting pass, cutting all links with weights less than or equal to k_i and return to Step 1.

Step 8: Select the largest previously unselected sub-graph (or sub-graphs if there are more than one of the same size) from the list compiled in Step 6 according to the following rules. Ignore all sub-graphs that intersect previously selected sub-graphs.

- a. If a largest sub-graph exists, select it.
- b. If more than one large sub-graph exists (tie), select all which have no intersections with other members of the selection group.
- c. If intersections exist, but the sub-graphs do not intersect all other sub-graphs, select the combination that has the greatest number of non-intersecting members.
- d. If the sub-graphs intersect all other sub-graphs or more than one combination of non-intersecting sub-graphs exists with the same numbers, select the sub-graph with the highest link density or the combination of sub-graphs with the highest average link density.
- e. If link densities are the same for the sub-graphs or combinations, select the sub-graph or combination with the lowest vertex weight sum.

(Note: The reason why the sub-graphs or combinations with higher vertex weight sums will not be selected is that they contain vertices which have more links/weight than other unselected vertices. If the sub-graphs or combinations with higher vertex weight sums were selected first, then the

Appendix G Graph Decomposition

size, i.e. the number of nodes, of the rest of the sub-graphs would be too small.)

- f. If vertex weight sums are the same for the sub-graphs or combinations, select the sub-graph or combination that was first discovered in the decomposition pass.

Step 9: Remove the selected sub-graph (or sub-graphs) from the graph and make it a block of the final partition. If the selected sub-graph is larger than previously established sub-graph blocks, then remove the smaller blocks from the partition and restore them to the graph.

Step 10: Examine the remaining vertices in the graph. If only disconnected singleton or doubleton sub-graphs remain, go to Step 11. If larger structures are still present, return to Step 1.

Step 11: Use the largest sub-graph (or sub-graphs if more than one of the same size exists) found in partition as the first block of the non-disjunctive decomposition.

Step 12: From the list of Step 6, select and remove according to the following rules the largest remaining sub-graph (or sub-graphs) covering previously uncovered vertices. Ignore all sub-graphs that are proper sub-graphs of those previously selected.

- a. If only one largest sub-graph exists, select it.
- b. If a tie exists, select all sub-graphs which independently cover previously uncovered vertices.
- c. If the same previously uncovered vertex or vertices are covered by more than one sub-graph, select the sub-graph which covers additional previously uncovered vertices.

Appendix G Graph Decomposition

- d. If the sub-graphs covering the same previously uncovered vertex or vertices cover the same number of additional previously uncovered vertices, select the sub-graph with the greatest link weight sum for the links connecting previously uncovered vertex (or vertices) and previously covered vertices (excluding the links among previously covered vertices).
- e. If the link weight sums are the same for the covering sub-graphs, select the sub-graph with a vertex (or vertices) having the greatest number of links.
- f. If the number of links from the vertex (or vertices) to the covering sub-graphs is the same, select the sub-graph that has the highest link density.
- g. If the link densities are the same for the covering sub-graphs, select the sub-graph with the lowest vertex weight sum.
- h. If the vertex weight sums are the same for covering sub-graphs, select the sub-graph that was first discovered in the partition process.

Step 13: Examine the remaining graph structure. If all vertices are covered, the decomposition is complete. Otherwise return to Step 12.

Some steps need to be explained in more detail for better understanding. In Step 6, a set of temporary sub-graphs is obtained. Each of the sub-graphs is respectively stored in a string array in which each element stands for a vertex. The temporary sub-graphs should be compared to those previously saved by comparing elements in two string arrays which store the sub-graphs.

In step 7, the cutting function value is computed. Cutting function is important in the algorithm and is not easy to understand at first sight because of its difficult expression. The complexity of expression for the cutting function results from forcing a continual increase on the value of the function. The reason for this is that most of the algorithm's activities are organized within the cutting pass loops. With each

Appendix G Graph Decomposition

successive cut there is a high probability that new calculations will again produce a structure with edge weights through the range from zero to the value of the cutting function. If the minimum weight criterion alone is used, the number of cutting passes required increases significantly. Forcing an increase in the value of cutting function for each pass would save time without accuracy loss. Arriving at Step 8 means one or more cutting passes have been done and finally a set of sub-graph blocks has been obtained. The goal of this step is to select the biggest one from the sub-graph blocks. Six rules are employed to make the selection.

The decomposition is purely topological. A $n \times n$ adjacency vertex matrix representing connection relationship is set up as the input table. The average connection ratio value of the whole graph is calculated as the initial value to control the decomposition process. If the decomposition result is not satisfied, this control value can be modified step by step to get the result. The size of modified step can be set as 0.1, 0.05 and 0.005 as a trial value. In each block, the manifold vertices should have degrees no smaller than 3 in the original graph. Each block would correspond to a real model.

The correct decomposition result would be produced for most of the pyramidal cases, even if the non-manifold vertices are not given before the decomposition. However, to handle drawings with more vertices as illustrated in Figure 6.9, the control value and its modified step are need to be adjusted carefully to produce the desired result. Therefore, a further and deeper investigation on graph decomposition is needed.