This is to certify that the

thesis entitled

Experiments in 3D CAD-Based
Inspection Using Range Images

presented by

Timothy S. Newman

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Computer Science

Anil Kumar Jain
Major professor

Date August 6, 1993

DATE DUE	DATE DUE	DATE DUE
~~MAY 2 3 1995~~ ~~[illegible]~~		
JUL 1 8 1999 8 02		

EXPERIME

in

EXPERIMENTS IN 3D CAD-BASED INSPECTION USING RANGE IMAGES

By

Timothy Scott Newman

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science Department

1993

EXPERIN

An experimen

objects that have b

automation of visu

machine vision sy

inspection operati

developed. This

images. The speci

quadric surfaces a

(range image) and

Our goal has b

a CAD model is

adaptation to insp

general-purpose i

process, sensor ca

ABSTRACT

EXPERIMENTS IN 3D CAD-BASED INSPECTION USING RANGE IMAGES

By

Timothy Scott Newman

An experimental system for automated visual inspection of three-dimensional industrial objects that have been designed with a commercial CAD system is presented. Historically, automation of visual inspection of 3D objects has been a difficult process. Special-purpose machine vision systems have been developed for some specific assembly guidance and inspection operations, but very few general-purpose visual inspection systems have been developed. This dissertation presents a system for inspection of castings using range images. The specific problem addressed is to ascertain if a 3D object consisting of planar or quadric surfaces and of known identity satisfies acceptability criteria, given its depth map (range image) and CAD model.

Our goal has been to create a system with inspection capabilities for any object for which a CAD model is available. Model-based inspection systems are flexible, allowing easy adaptation to inspection of a new object. In this dissertation, we demonstrate that flexible general-purpose inspection is possible when the problem domain—i.e., the manufacturing process, sensor capabilities, and object types—is understood. The problem domain consists

of range images of small

A suite of model-dri
classification problem i
classification technique
allow the detection of
pits and short pours) ar

The sensor limitat
of the casting process
image segmentation
Several techniques f
Hausdorff matching.
their suitability to the
presented and then in

A gross defect d
inspection is present
features and dimensi
CAD modeler is use
is not a complete in
demonstrate the feasib
of the techniques pres
implemented in the C

of range images of small- to medium-sized castings for which there is a CAD model.

A suite of model-driven 3D surface fitting algorithms that are applicable to the surface classification problem in object recognition is presented. Through extensions to these classification techniques, surface shape inspections are also carried out. These inspections allow the detection of common defects in the casting process, including insufficient (e.g., pits and short pours) and excess (e.g., drops) material defects, and warps and mismatches.

The sensor limitations and their impact on inspection are investigated. The effects of the casting process on the inspection tasks are also examined. Furthermore, a range image segmentation scheme is presented to allow more reliable casting surface extraction. Several techniques for object localization, including the Extended Gaussian Image, the Hausdorff matching, and, particularly, the interpretation tree search, are investigated for their suitability to the inspection task. Parallel algorithms for interpretation tree search are presented and then implemented and analyzed on a MIMD machine.

A gross defect detection scheme based on random template matching for in-process inspection is presented. The system also has capabilities to inspect several high-level features and dimensions. Information extracted from models built with a commmercial CAD modeler is used throughout the system. The system presented in this dissertation is not a complete inspection system, but rather is an experimental system designed to demonstrate the feasibility of general-purpose inspection of castings in range images. Many of the techniques presented are also applicable in other inspection realms. The system was implemented in the C language and tested on a large number of real and synthetic images.

Now to the King eternal, immortal, invisible,
To God alone who is wise. (I Timothy 1:17)

I would like to ack
Northrop Research and
provided partial supp
to Michigan State Un
gracious and helpful t
Laboratory over the y
rest of the engineering
sharing their knowled
supplying sample cas
Advanced Computing
Argonne National Lab

I also wish to than
years of graduate stud
Anil K. Jain, has been
to sharpen my focus o
have developed a great
University and feel qu
have each been helpf
has provided practical
delving into computer
computing facilities of
me appreciate many is
Richard Enbody has al
helped nurture my kno
has helped me apprecia
John Forsyth, Dr. Mih
Though not on my com
over the years.

Many fellow stude

ACKNOWLEDGMENTS

I would like to acknowledge the financial and practical supporters of this research. The Northrop Research and Technology Center (NRTC) and the National Science Foundation provided partial support for this work (the NSF through infrastructure grant CDA-8806599 to Michigan State University.) Dr. H. R. Keshavan of NRTC has been particularly gracious and helpful to me and to the Pattern Recognition and Image Processing (PRIP) Laboratory over the years. I also wish to thank Robert Huber, Jim Bujouvides, and the rest of the engineering staff at General Motors Saginaw Malleable Iron Plant for graciously sharing their knowledge about the casting process and common casting defects, and for supplying sample castings for the experiments in this thesis. We also acknowledge the Advanced Computing Research Facility, Mathematics and Computer Science Division, Argonne National Laboratory, for providing access to the BBN TC-2000.

I also wish to thank all the individuals who have assisted me in some way during my years of graduate study. In particular, the professional advice of my thesis director, Dr. Anil K. Jain, has been very beneficial to my development as a researcher and has helped to sharpen my focus on the many issues involved in object recognition and inspection. I have developed a great respect for Dr. Jain's knowledge during my stay at Michigan State University and feel quite indebted to him. The other members of my advisory committee have each been helpful in various aspects of my development. Dr. George Stockman has provided practical insight into manufacturing processes and has spent many hours delving into computer vision problems with me. Dr. Eric Goodman has provided the computing facilities of the Case Center for Computer-Aided Engineering and has helped me appreciate many issues in geometric modeling and computer-aided manufacturing. Dr. Richard Enbody has always had an open door, been a welcome sports teammate, and has helped nurture my knowledge and enthusiasm for parallel computation. Dr. James Zacks has helped me appreciate the wonders of animate vision processes. I also want to thank Dr. John Forsyth, Dr. Mihran Tuceryan, Dr. John Weng, and, especially, Dr. Richard Dubes. Though not on my committee, these faculty members have provided a great deal of support over the years.

Many fellow students and colleagues from the MSU PRIP Laboratory have provided

moral support and help
lab has been quite cor
for his assistance both
Dame and Washington
Dorai, Dr. Farshid Fa
Dr. Narayan Raja hav
provided LaTeX and s
of Philippe Ballard, Dr
Jonathan Courtney, M
Fukui, Dr. Bilge Gun
Dr. Sateesha Nadabar
Dr. Steve Walsh, and
and prayers of my par

moral support and helped refine many of my investigations. The environment of the PRIP lab has been quite conducive to research. Dr. Patrick Flynn deserves special mention for his assistance both while he was at Michigan State and later when he was at Notre Dame and Washington State Universities. Fellow students Sushil Bhattacharjee, Chitra Dorai, Dr. Farshid Farrokhnia, Dr. Greg Lee, Jian-Chang Mao, Dr. Joseph Miller, and Dr. Narayan Raja have all been especially helpful. John Lees and J. Daniel Smith have provided LaTeX and systems consultation. I also appreciate the assistance and friendship of Philippe Ballard, Dr. C.C. Chen, Jinlong J. Chen, Dr. Sei-Wang Chen, David Chesney, Jonathan Courtney, Marie-Pierre Dubuisson, Hansye Dulimarta, Jonathan Engelsma, Yukio Fukui, Dr. Bilge Gunsel, Dr. Salim Hiziroglu, Sally Howden, Qian Huang, David Marks, Dr. Sateesha Nadabar, Sharath Pankanti, Dan Swets, Dr. Torfinn Taxt, Dr. Deb Trytten, Dr. Steve Walsh, and Don Yuvan. Finally, I am deeply appreciative of the support, love, and prayers of my parents and family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

As marketplaces have become increasingly global in their scope, many industries have experienced heightened concerns about their competitiveness. Domestic manufacturers cannot rest content in their once-captive home markets, for example, because consumers are always eager to find the best product at the least cost among a larger number of choices. Traditionally, technological improvement has been one of the most effective strategies for maintaining or increasing a company's market share. Manufacturers in the United States, and especially those in Japan and Western Europe, have used robotics and automation on the factory floor to improve product quality and efficiency and to reduce costs. Robotics, coupled with quality assurance methods, can also reduce product recalls and warranty and maintenance costs [113].

For robots to become more widespread in manufacturing, however, they must be provided with some visual abilities. Human workers display great flexibility on an assembly line, partly because their visual system allows them to rapidly recognize and handle unexpected or unusual situations. Some robots already use vision systems to perform "intelligent" tasks in a variety of application domains. For example, machine vision is used to help position welding and painting robots [20, 63, 141, 153, 154, 178], for guiding insertion operations in assembly operations [141, 178], and for decoration of foodstuffs [141].

One area that has shown some resistance to automation is visual inspection of parts.

Inspection is a task t
of vision-based ins
performed for very
which can be easily
of the existing auto
systems (e.g., [111
highly constrained.
to a class of object
product.

The ultimate a
solid object for w
of CAD models m
is easily adaptable
research, inspectio
however. This diss
models, object loc
techniques for ins
Parallel algorithm

A general-pur
defects. These in
integrity inspectio
to perform these
surface shape insp

1.1 Definit

Inspection is the p

Inspection is a task that requires flexibility in addition to visual abilities. As such, automation of vision-based inspection tasks has been a slow process. Automated inspection has been performed for very specialized applications, but there is no automated inspection system which can be easily adapted for inspecting a wide variety of products. To our knowledge, all of the existing automated inspection systems, with the exception of very few experimental systems (e.g., [111]), have been designed to inspect a single object or part whose position is highly constrained. This dissertation presents automated inspection algorithms applicable to a class of objects, rather than a specialized system only usable for inspection of a single product.

The ultimate aim of this research is to be able to inspect any three-dimensional (3D) solid object for which a Computer Aided Design (CAD) model is available. The use of CAD models makes the inspection system flexible; a system that uses CAD models is easily adaptable to inspection of new objects or even new classes of objects. In this research, inspection is restricted to 3D iron castings composed of planar or quadric surfaces, however. This dissertation addresses issues of model representation, feature extraction from models, object localization, fusion of data from multiple viewpoints, efficient matching and techniques for inspection of surface shape, dimensional tolerance, and assembly integrity. Parallel algorithms for matching and localization are also presented and analyzed.

A general-purpose industrial inspection system needs to detect three major types of defects. These include part dimensioning (metrology), surface inspection, and assembly integrity inspection. The goal of this research is to present a system that includes techniques to perform these three inspection functions. The experiments focus on metrology and surface shape inspection, however.

1.1 Definition of Inspection

Inspection is the process of determining if a product (also referred to as a part, object, or item,

both in this disserta
81, 97, 99]. Inspe
assembly integrity.
is distinguished fro
of specific operatio
contrast, consists o
usually is not neces
priori. Thus, inspec

Some of the pap
perform either recog
presented an Hough
inspection technique

Inspection shoul
tasks. Guidance and
feedback to a robot [
as an inspection syst
control system. Partr
position has been de
performed.

1.2 The Cast

In this research, the
General Motors (GM
this dissertation are a
While it might seem
independent of the p

both in this dissertation and in the literature) deviates from a given set of specifications [73, 81, 97, 99]. Inspection usually involves measurement of specific part features such as assembly integrity, surface finish, geometric dimensions, etc. It is a quality control task, but is distinguished from *testing* tasks, such as stress analysis, that involve active examination of specific operational functions of the product [99]. Part identification or *recognition*, by contrast, consists of the positive identification of an object. Identification of the object usually is not necessary for the inspection task; identity is usually assumed to be known *a priori*. Thus, inspection, testing, and recognition are distinct but related tasks.

Some of the papers in the literature which claim to present inspection methods actually perform either recognition or segmentation. Ker and Kengskool [98], for example, have presented an Hough Transform-based edge segmentation method that they claimed as an inspection technique.

Inspection should also be distinguished from guidance and control (i.e., *localization*) tasks. Guidance and control consist of measuring the relative position of objects to provide feedback to a robot [117]. The Partracker system by Automation, Inc., which was reported as an inspection system in the literature [141], is actually an example of a guidance and control system. Partracker locates parts whose position is not known precisely. After part position has been determined, Partracker allows a welding or assembly operation to be performed.

1.2 The Casting Inspection Problem

In this research, the domain of inspection is restricted to iron castings produced by the General Motors (GM) Saginaw Malleable Iron (SMI) Plant. The techniques presented in this dissertation are aimed at detecting defects that commonly occur in the casting process. While it might seem ideal to present inspection schemes that can function well totally independent of the problem domain, in practice this is not feasible. In a given application, certain defects seld

detected. The choice

method. Thus the ap

Although the pr

veloped here are ap

restrictions on which

range sensor rather t

research and its limi

1.2.1 The Castin

Casting is the process

usually made of sand

the top part of the m

molds also contain c

This allows the creat

and drag are pressed

casting.

Example Images

An intensity and a ra

shown in Figure 1.1.

viewpoints. The cast

356 (referred to as g

near the gate stub (th

of the part). The part

boundary at a positio

overlaid on the range

certain defects seldom, if ever, occur, while other defects are very important and must be detected. The choice of a defect detection scheme is also tightly coupled with the sensing method. Thus the approach to inspection in this dissertation is domain-specific.

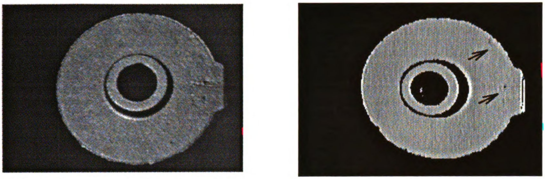
Although the problem domain is restricted to castings, the inspection techniques developed here are applicable to objects that contain a wide class of surfaces. Most of the restrictions on which surfaces and defects can be inspected were limited by the available range sensor rather than the object classes. Details about the range sensor used in this research and its limitations are presented in Chapter 4.

1.2.1 The Casting Process

Casting is the process of forming an object by pouring molten metal into a mold. The mold, usually made of sand, consists of a "top" half and a "bottom" half. The *cope* is the name for the top part of the mold while the *drag* is the term for the bottom part of the mold. Some molds also contain *cores* that allow the pouring of castings which have hollow centers. This allows the creation of various housings and some gear and pulley blanks. The cope and drag are pressed together and then molten metal is poured into the mold, to create the casting.

Example Images

An intensity and a range image of one of the defective iron castings from GM SMI are shown in Figure 1.1. These images were collected from similar, although not identical, viewpoints. The casting in the images is a gear blank whose part identification number is 356 (referred to as gear blank 356 henceforth). This casting contains a small surface pit near the gate stub (the gate is the rectangular obtrusion from the generally circular border of the part). The part also contains a small amount of excess material near the outer circular boundary at a position of about 35 degrees counter-clockwise from the pit. Arrows are overlaid on the range image to mark the position of both of the defects. Range images of



(a) Intensity image of defective casting.

(b) Pseudo-intensity rendering of depth map of defective casting.

Figure 1.1. Intensity and range images of a defective gear blank casting.

two other castings, both defect-free, are shown in Figures 1.2 and 1.3.

1.2.2 Casting Defects

This section discusses some of the common defects encountered in the casting process. Most of the material presented in the remainder of this section is based upon discussions with the staff at GM SMI, especially with Mr. Robert Huber [80]. Several handbooks of casting defects were also consulted [1, 2].

Holes and Cracks

Two of the most common surface defects in castings are *holes* and *cracks*. These defects are also two of the more critical defects so their detection is important. As discussed above, surface defects weaken the part by reducing its ability to withstand mechanical and thermal loads [62].

Several classes of holes exist in castings, including gas (or blow) holes, slag holes, and sand holes. Gas holes are caused by pockets of gas becoming trapped in the mold. Inclusions, or slag holes, are caused by impurities in the molten metal. This is usually caused by foreign material, such as sand, becoming embedded in the metal during casting.

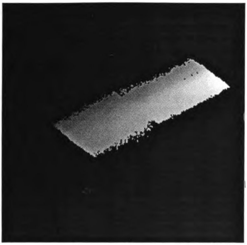


Figure 1.2. Depth map of a cylindrical casting. (Intensity coding of depth.)

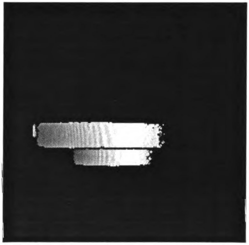


Figure 1.3. Pseudo-intensity rendering of depth map of gear blank 588x casting.

This foreign material usually washes off during cleaning, however, leaving small pits in the casting surface. An example of an inclusion defect, from [1], is shown in Figure 1.7. Sand holes are most often caused by incorrect moisture in the molding sand. This causes moisture from the sand to condense on the surface of the metal during cooling, leaving pits behind after the metal has cooled and the moisture evaporated. The classes of hole defects all appear as pits or "bubbles" on the surface of the casting. There is some overlap in the "symptoms" of these defects, and many experts disagree on the exact classification of defects. In this dissertation, all hole-type defects are grouped together into a single class. We will refer to this class with interchangeable terms such as "pit," "blow hole," or "hole." A pit defect is shown in Figure 1.1.

Cracks are often caused by incorrect rates of cooling and appear as hairline fractures in the surface of the casting. Cracks are very common defects in the casting process. Currently, GM detects cracks using a system called MAGSIGHT. This system induces magnetic flux on the castings and then illuminates the castings with fluorescent light. A human operator views the castings ar

the magnetized castin

Insufficient Mater

Another common de
metal being poured
as misshapen surface
short defects as *misr*
lacks completeness,
the mold. A misrun
extremity of the mold
There appears to be
refer to all of them as
are shown in Figures
to poured short defec
mold. This defect is

Excess Material D

Excess material def
as *drop* defects, are
Figure 1.6. Drop d
the casting. Runout
surfaces of the casti
this dissertation, no

'MAGSIGHT is a m

views the castings and removes defective parts.* MAGSIGHT uses four cameras to view the magnetized castings.

Insufficient Material Defects

Another common defect is a *poured short* defect. This defect is caused by insufficient metal being poured into a mold. It is usually manifested as a gross feature defect such as misshapen surfaces or missing features. Foundrymen sometimes classify some poured short defects as *misruns* or *shrink* defects. Technically, a short pour is any casting that lacks completeness, usually caused by an insufficient amount of metal being poured into the mold. A misrun is a casting that is missing a feature because metal did not fill an extremity of the mold while a shrink defect is any depression from the true mold surface. There appears to be a great deal of overlap in the definition of these three defects, so we refer to all of them as poured short defects. Two examples of poured short defects from [1] are shown in Figures 1.4 and 1.5. *Runout* defects may also cause defects that appear similar to poured short defects, although the runout defect is caused by metal "running out" of the mold. This defect is usually the result of a mismatch between the cope and drag.

Excess Material Defects

Excess material defects caused by sand dropping away from the mold, often classified as *drop* defects, are also common. An example of a drop defect from [1] is shown in Figure 1.6. Drop defects usually cause excess material to be deposited on the surface of the casting. Runout defects, which often are manifested as insufficient material on some surfaces of the castings, usually also cause excess material on other areas of the casting. In this dissertation, no distinction is made between the different excess material defects.

*MAGSIGHT is a magnetic particle testing (MPT) method. MPT is discussed in section 2.2.3.

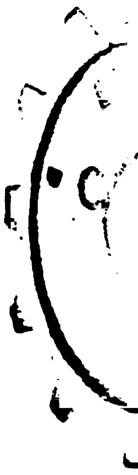


Figure 1.4. Exa
(misrun) defect;
[1], used by perr

Figure 1.6. An
From [1], used

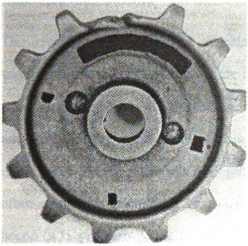


Figure 1.4. Example of a poured short (misrun) defect; a tooth is missing. From [1], used by permission.

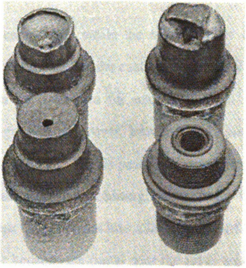


Figure 1.5. An example of a poured short (shrink) defect. From [1], used by permission.

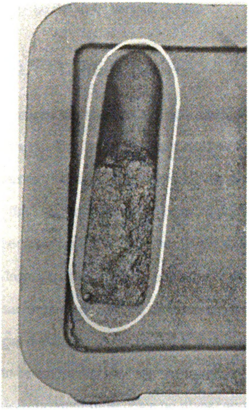


Figure 1.6. An example of a drop defect. From [1], used by permission.

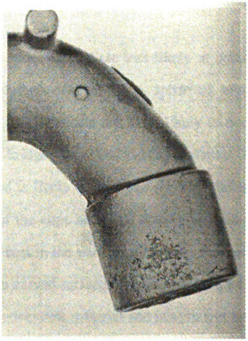


Figure 1.7. An example of an inclusion defect. From [1], used by permission.

Material Allowances

The tolerance for insufficient material is typically 0.06" while the tolerance for excess material is about 0.12". Castings are usually machined after the casting process, allowing small defects (especially defects smaller than the tolerance for excess material) to be removed. Hence, most small defects can be ignored. However, human inspectors often reject castings with small pits of depths as little as 0.02". This is because a small pit defect often cannot be completely removed during subsequent machining; many times the pit becomes larger during machining, either because the pit opens into a larger sub-surface pit (that is not visible through visual inspection) or because the metal near the pit is weak and breaks away during machining. Cracks may be as small as 0.01" deep with a similar width, or they may be deeper fissures of a barely visible width. Cracks occur quite commonly and may be the most serious defect.

1.2.3 Unlikely Defects

Because of the nature of the casting process, some defects are less likely to occur than others. For instance, there tend to be fewer defects on cylindrical, spherical, and other curved surfaces than there are on planar surfaces. Defects are most likely to occur on "flat" surfaces or on the blended surfaces that bridge planar and curved surfaces. This is because sand that falls out of the casting mold is likely to fall onto a horizontal surface during casting. The x-y (horizontal) planes of the cope and drag usually contain planar surfaces while curved surfaces are most often cast in the vertical direction. Furthermore, gas defects are more probable on planar than on curved surfaces because gas is more likely to be trapped at the "top" of the mold. Thus, both excess material and insufficient material defects occur most often on planar surfaces or on the filleted surfaces that blend curved and planar surfaces of castings.

Another uncommon defect is a "shift" within a cope or drag section of the casting. It

is often the c
the "bottom"
The acceptab
tolerance def
be observed w
"bottom" of t
are unlikely t
casting.

A defect s
because the p
section of the
runout on sa
using a coord
Scanning wi
scan commo
tolerances se
as important

1.2.4 Ins

GM's Sagina
are currently
inspected. Ap

In the earl
CONSIGHT t
casting orient
system was re
work properly

is often the case that there is a *mismatch* or shift between the cope and drag that causes the "bottom" part of the object to be shifted relative to the "top" of the object, however. The acceptable limit on such shifts is 0.03". A mismatch causes most of the dimensional tolerance defects in the casting process. Most of the dimensional tolerance defects would be observed when measuring between one point from the "top" and another point from the "bottom" of the casting. But shifts entirely within the cope or drag are unlikely. Thus, there are unlikely to be any out-of-tolerance dimension measurements on the same "side" of the casting.

A defect such as deviation from circularity is one example of a defect that is uncommon because the plane that contains a circle is usually entirely within either the cope or the drag section of the casting. Circularity defects are currently detected by measuring total indicator runout on samples collected approximately once per week. Other tolerances are checked using a coordinate measuring machine (CMM) on samples collected about once per day. Scanning with a coordinate measuring machine is time-consuming, however. A CMM scan commonly takes about two hours. The checks for circularity and other dimensional tolerances serve primarily to check the tooling of the casting process; such a defect is not as important as the others because it would generally not occur in isolated instances.

1.2.4 Inspection in the Foundry

GM's Saginaw Malleable Iron Plant creates over 100 different castings. All of the castings are currently sorted and inspected by hand. Typically, about 285 castings per hour must be inspected. Approximately 5% of the castings contain unsalvageable scrap (i.e., defects).

In the early 1980's, GM installed an intensity-image based machine vision system called CONSIGHT to automatically inspect and sort castings. It had very general allowances for casting orientation but required that there be no overlap between castings. This vision system was removed after about one year of use, however, primarily because it failed to work properly for parts that were definitionally non-defective yet still deviating from the norm. Ultimately,

that it had inspecte

the castings, inclu

always included i

small differences i

parts that were d

only a $\frac{1}{8}$" differe

functioning in th

There are se

For example, the

the final casting

this dissertation

at the foundry

is also the one

collected from

1.2.5 Insp

The casting i

and because

challenge fo

images and

The insp

from CAD

structure pa

mathematic

manufactur

quality. Th

norm. Ultimately, a human had to be stationed next to CONSIGHT to re-sort the parts that it had inspected. One reason for CONSIGHT's failure was the natural roughness of the castings, including the presence of additional features such as gate stubs that were not always included in CONSIGHT's model base. Many of the parts being sorted had very small differences in dimensions, making it difficult for CONSIGHT to distinguish between parts that were defective and parts that were simply different. Some of the parts had only a $\frac{1}{8}$" difference in dimensions, for example. CONSIGHT also experienced problems functioning in the harsh environment of the factory floor.

There are several places in the foundry where automated inspection would be useful. For example, the molds could be inspected after construction and again just before pouring, the final casting products could be inspected, and hot castings could also be inspected. In this dissertation, we focus on inspection of cold castings in the finishing area. The personnel at the foundry believed that the most natural place for inspection is in the finishing area. It is also the one part of the inspection process that can be developed off-site using samples collected from the foundry.

1.2.5 Inspection Problems Studied in this Thesis

The casting inspection problem is quite challenging because of the nature of the defects and because the castings contain natural roughness caused by the grain of the sand. The challenge for this work is to robustly segment, classify, and identify the castings in (range) images and to determine the nature of any defects present.

The inspection system in this dissertation also relies on model information extracted from CAD models of castings. CAD packages allow a designer to manipulate shape and structure parameters and observe their influence on part geometry. CAD can also provide a mathematical description of object shape. They can also be extended to contain additional manufacturing information, including dimensional tolerances and surface finish type and quality. The CAD model therefore presents a centralized object description valuable for

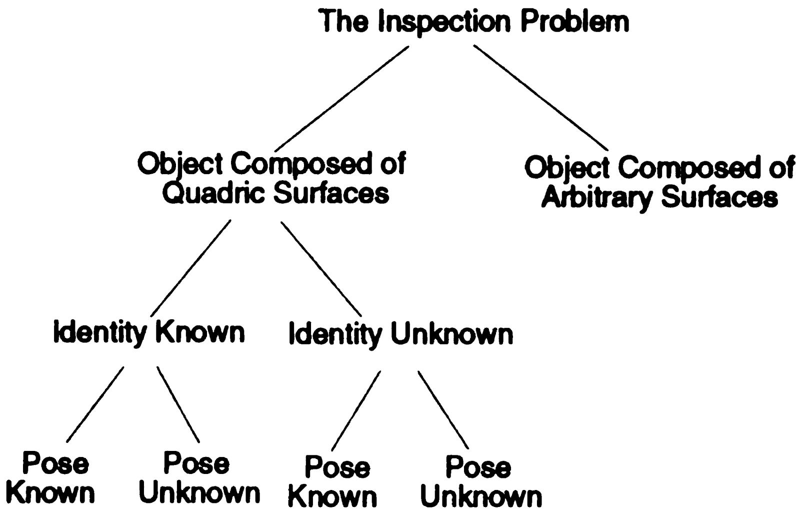


Figure 1.8. Dichotomy of Inspection Problems.

interrogation for inspection. Chapter 3 discusses the advantages of CAD models in depth. The reliance on CAD models in this work also increases the potential to apply the inspection methods to other industrial inspection problems.

The work presented here inspects the castings using range images because 3D surface information is explicitly represented in range images. This also makes our methods generally applicable to many other inspection problems. The advantages of range images are discussed in more detail in Chapter 4.

Figure 1.8 shows a dichotomy of the various inspection problems. In the diagram, the unlabeled subtree under "Arbitrary Surfaces" is identical to the subtree under "Quadric Surfaces." At the highest level, inspection tasks can be distinguished by the surface types that compose the object boundary. The second level of the tree indicates if inspection requires solution of the object recognition problem. Finally, the lowest level of the tree indicates if the object's pose is known, perhaps through control of fixtures, or unknown. In this dissertation, we focus mostly on the leftmost branch of the tree; known objects with planar or quadric surfaces that are in a given pose are inspected. In Chapter 6, techniques for pose and identity determination are presented, however.

1.3 The

In this thesis
castings usin
tolerances, g
regions, and s
presented her
have been tes
knowledge, th
on a set of re

The contr
itations of th
investigated.
image segmen
ings. A suite
quick and ac
fitting routine
minor segmen
object recogni
matching to st
that has been
feasibility of p
gorithms for g
surface shape,

A benefit o
of objects rath
applied to man

1.3 Thesis Contributions

In this thesis, we present a CAD-based system for general-purpose visual inspection of castings using range images. The techniques are useful for inspection of dimensional tolerances, gross assembly integrity, surface shape of cylindrical, planar, and spherical regions, and surface defects such as pits, excess material, short pours, etc. The techniques presented here are also able to detect some crack defects. Furthermore, the techniques have been tested on a set of real objects manufactured by an automotive company. To our knowledge, this is one of the first general-purpose inspection methods to have been tested on a set of real objects.

The contributions of this work include the following. First, the capabilities and limitations of the 100X range sensor and the requirements for the model representation are investigated. We also present an enhancement of the Flynn-Hoffman-Jain [55, 75] range image segmentation scheme that produces reasonable segmentations of the images of castings. A suite of techniques for model-driven quadric surface fitting is presented for the quick and accurate classification of regions in range images. Extensions to the surface fitting routines are presented that allow robust surface fitting in the presence of defects and minor segmentation errors. We investigate several techniques for pose localization and object recognition—including the Extended Gaussian Image, Interpretation Tree Search, matching to stored views, and the Hausdorf matching—and present a tree search technique that has been implemented and analyzed on a MIMD computer, thus demonstrating the feasibility of parallel computation for high-level vision applications. We also present algorithms for gross defect detection, detection of common casting defects, inspection of surface shape, and verification of some feature attributes and dimensional tolerances.

A benefit of our methodology is that the techniques presented are applicable to a class of objects rather than to a specific single object. Our techniques have been successfully applied to many range images of castings, thus demonstrating their utility. The work in this dissertat

visual proces

castings in ra

recognition ar

visual processi

1.4 Thes

The dissertation

and discusses m

the literature. Cl

the representation

range images and

presents the sensin

extraction, segmen

object localization.

defects. In this rese

that pass the gross fo

the success of opera

casting surface defe

inspection of feature

Chapter 9. The disse

A simplified blo

ure 1.9. The boxes

in which each step

provided at the begin

this dissertation also makes use of information from the CAD model in various steps of visual processing. Although the techniques presented here are specific to the domain of castings in range images, many of the techniques have general applicability to 3D object recognition and inspection due to the reliance on general models throughout the steps of visual processing.

1.4 Thesis Organization

The dissertation is organized as follows. Chapter 2 presents typical inspection problems and discusses many of the automated visual inspection systems that have been reported in the literature. Chapter 3 discusses CAD models and data representation issues, including the representations specific to this research. Chapter 4 reviews the techniques for acquiring range images and motivates the need for the use of range images for inspection. It also presents the sensing arrangement used in this thesis. Chapter 5 discusses surface feature extraction, segmentation, and surface classification. Chapter 6 presents techniques for object localization. Chapter 7 explains the technique proposed for inspection of gross defects. In this research, the gross feature inspection is the first stage of inspection. Objects that pass the gross feature inspection are then inspected for common casting defects and for the success of operations of interest. Chapter 8 contains the techniques for many common casting surface defect inspections, which can be called surface shape inspection. The inspection of features or operations of interest and dimensional tolerances is discussed in Chapter 9. The dissertation is concluded in Chapter 10.

A simplified block diagram of the steps in our inspection system is presented in Figure 1.9. The boxes in the figure indicate the order of the inspection steps and the chapter in which each step is discussed. An expanded view of the box labeled "Inspection" is provided at the beginning of Chapter 7.

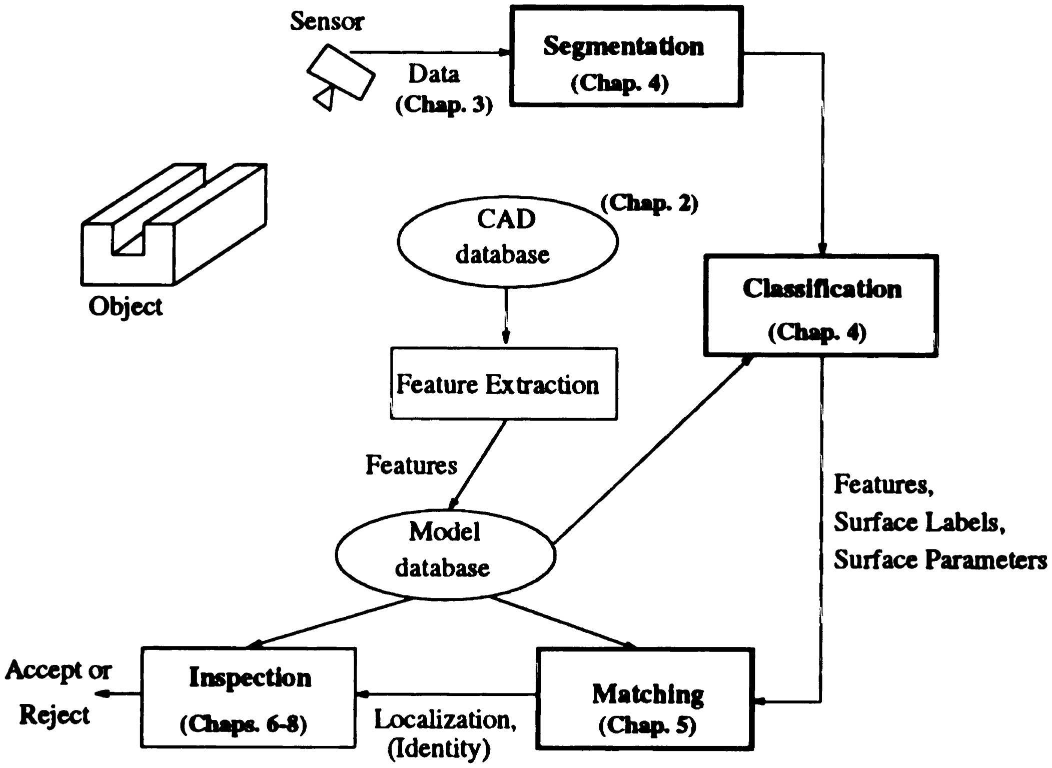


Figure 1.9. Block Diagram of Inspection System.

CHAPT

Backgro

In this chapter, we
of automated visu
visual inspection s

2.1 A Taxo

In this section, we

2.1.1 Inspecti

There are three ge
are *input* (also cal
spection [44, 63, 9
tion [44, 97, 159, 16
to determine if thei
a sufficient amount
of the output of an i
at a stage were perf
is in control or if too

CHAPTER 2

Background on Inspection

In this chapter, we discuss the stages of inspection and motivate the need for and feasibility of automated visual inspection. We also present an overview of many of the automated visual inspection systems that have been reported in the literature.

2.1 A Taxonomy of Inspection Problems

In this section, we present a taxonomy of the inspection problems.

2.1.1 Inspection Stages

There are three generally accepted application areas (or stages) for inspection. These are *input* (also called *receiving* or *incoming*) *inspection* [44, 97, 166, 185], *process inspection* [44, 63, 94, 97, 159, 166, 178, 185, 194], and *output* (or, *product*) *inspection* [44, 97, 159, 166, 185, 194]. Receiving inspection is the examination of raw materials to determine if their quality is acceptable for use. It also involves determining if there is a sufficient amount of material for use in assembly. Process inspection is the examination of the output of an intermediate work stage; it is useful for determining if the operations at a stage were performed within specified tolerances and whether the assembly process is in control or if tools are worn or broken. Process inspection allows fine adjustments to

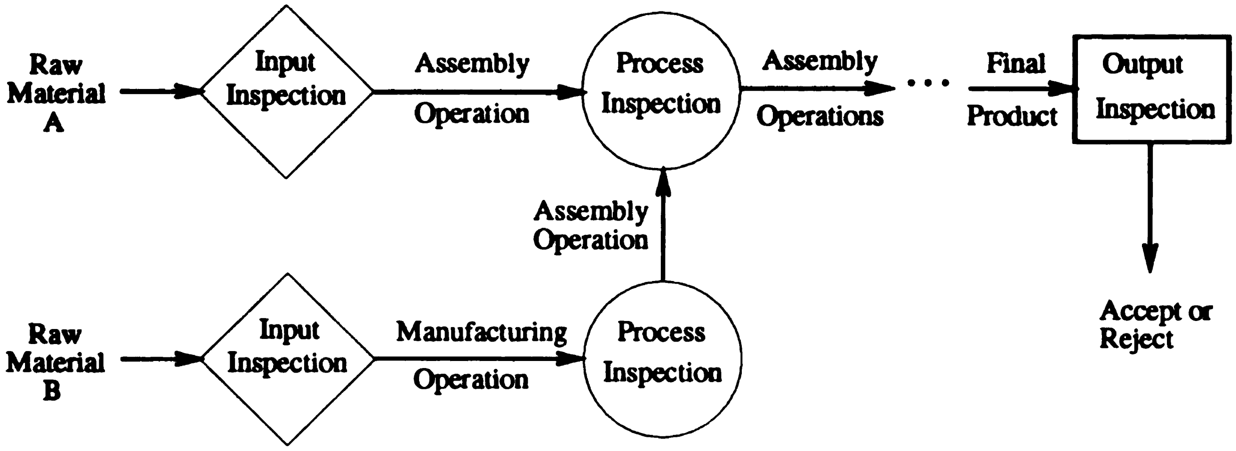


Figure 2.1. Stages of inspection in a production sequence.

be made for tool wear and helps prevent nonconforming parts or material from being used in a later production stage. The advantages of process inspection are discussed further in Section 2.1.3. Output inspection is the final exhaustive inspection of a product at the end of all assembly or manufacturing stages to determine the product's acceptability. Output inspection is also often used to collect statistical information to discover long-term trends in the assembly process, for instance to find that tools need to be replaced or that general maintenance may be required [166]. Figure 2.1 shows several stages of production and inspection.

2.1.2 Inspection Dichotomies

Wetherill [185] has presented several dichotomies of inspection. One dichotomy for batch inspection is *rectifying inspection* versus *acceptance inspection.* Rectifying inspection refers to sorting out bad products from the batch and adjusting (or replacing) those items. Acceptance inspection involves the labeling of batches of products into categories such as "accept," "reject," or, perhaps, "seconds" (e.g., products to be sold at a lower price).

Another dichotomy of inspection tasks is *inspection by attributes* versus *inspection by variables* [185]. Inspection by attributes classifies objects as either acceptable or defective.

Mair prefers the ter

attribute inspection

checking labels, exa

ishes), detecting crac

spurious and missing

of some feature, suc

are termed *quantitat*

tures and the symbo

chart in Figure 2.2, t

to indicate manufac

important part dime

Grading inspection

acceptability of fea

produce, for exampl

Lee [104] ident

detecting that "no

implicit inspection i

improper reflective

appear to overlap

involves recognizi

can easily recogniz

by explicit parame

open research prob

processing requiren

Mair prefers the term *qualitative inspection* for these visual inspections [109]. Typical attribute inspection tasks include qualitative and semiquantitative mensuration, such as checking labels, examining cosmetic and surface finish properties (such as colors or blemishes), detecting cracks and flaws, and verifying part integrity and completeness (including spurious and missing features) [109, 148]. Inspection by variables produces a measurement of some feature, such as the product's length or weight. These dimensional inspections are termed *quantitative inspection* by Mair [109]. Some of the quantitative geometric features and the symbols commonly used to describe them on blueprints are provided in the chart in Figure 2.2, taken from Kennedy et al. [97]. These symbols are used on blueprints to indicate manufacturing tolerances. Another inspection by variables task is measuring important part dimensions such as exterior wear, tool wear, and workpiece dimensions. *Grading inspection* is a hybrid classification that classifies products based on degree of acceptability of features, such as length or weight [185]. One example of this is grades of produce, for example, grade A versus grade AA eggs.

Lee [104] identifies an inspection task he terms *implicit inspection* which involves detecting that "novel but 'obviously' faulty items" are defective. Defects detected by implicit inspection include poor finishes, dents, scratches, incorrect color, wrong orientation, improper reflective properties, and other ill-defined faults. While these defects might appear to overlap with some of the inspection tasks defined above, implicit inspection involves recognizing an unanticipated defect. Human inspectors are quite flexible and can easily recognize novel defects. Many of these unanticipated defects are not described by explicit parameters, making automated detection difficult. Implicit inspection is an open research problem, calling for sophisticated representations of the object, task, and processing requirements [104].

Fo

Pr

Or
tat

Lo

Ru

Figure 2.2. Comm
Kennedy et al. [97]

2.1.3 Why Au

100% Inspectio

In industrial envir
a small-sized sam
quality characteris
was drawn. Some
tends to be more a
(where every prod
inconsistency. As
requires high level

However, in so

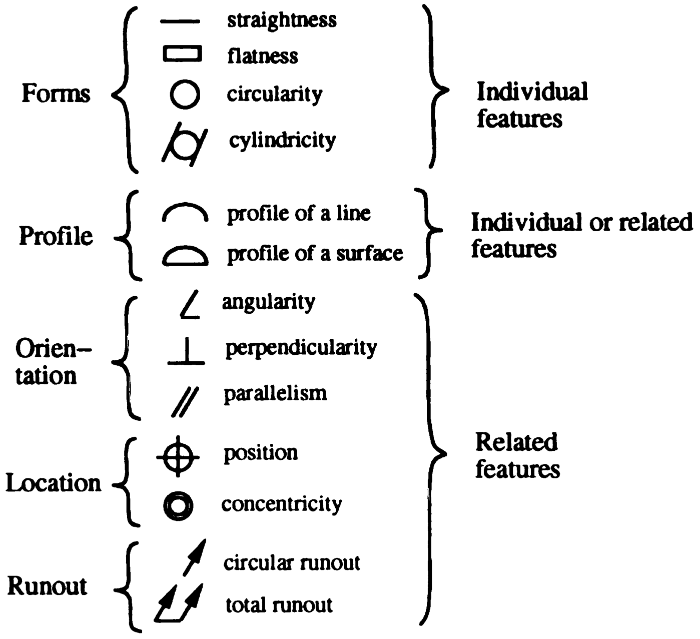


Figure 2.2. Common blueprint symbols for geometric characteristics and features from Kennedy et al. [97] .

2.1.3 Why Automated Inspection?

100% Inspection vs. Batch Inspection

In industrial environments, inspection has usually been performed by human inspectors on a small-sized sample from the lot or batch. In this modality, called batch inspection, the quality characteristics of the sample are generalized to the batch from which the sample was drawn. Some experiments have indicated that batch inspection by human inspectors tends to be more accurate than an inspection modality of 100% inspection of parts [185] (where every product in the lot is inspected), probably because of inspector fatigue and inconsistency. As a result, achieving 100% inspection using human inspectors typically requires high levels of redundancy, thus increasing the cost and time for inspection [45].

However, in some critical applications, such as aerospace and medicine, even a single

faulty pr
especially
also desir
Part suppl
of product
inventory t
terms of lo
products, ar
inspection i.

Samplin
from a large
inferential re
with samplin
is the risk ass
the extraction
the chance of
containing ma
consumer's ris
and market sha

General Bene

Automatic inspe
consistently. Th
quality to be pred
boring for human
account for 10% c
inspectors also of

faulty product is unacceptable. In other markets, notably those for consumer products, especially foodstuffs, products having aesthetic appeal sell better [74]. Many manufacturers also desire 100% inspection to enhance a product's competitiveness in the marketplace. Part suppliers to factories using just-in-time inventory practices are especially conscious of product quality since the recipient factories are generally unwilling to store and pay for inventory that requires testing before use [113]. The cost of poor quality components (in terms of lost sales) has been estimated to be between 5% and 10% of the total sales for most products, and approximately 20% of sales for manufactured goods [113]. Therefore, 100% inspection is desirable.

Sampling can be formally defined as the "process of selecting a representative set of data from a large number of measurements or observations (the population) in order to perform inferential reasoning about the population itself" [44]. Two types of risks are associated with sampling. These are known as *producer's risk* and *consumer's risk.* Producer's risk is the risk associated with a lot or batch of good quality that is mistakenly rejected due to the extraction of a random sample containing many defective parts. Consumer's risk is the chance of mistakenly accepting a lot of poor quality due to the extraction of a sample containing many good parts. There are many consequences associated with high levels of consumer's risk, although the end result of all of them is that the manufacturer's reputation and market share are endangered. Consumer's risk can be reduced through 100% inspection.

General Benefits of Automated Inspection

Automatic inspection is desirable because human inspectors do not always evaluate products consistently. The consistency associated with automated inspection should allow the level of quality to be predicted [94], however. Many inspection tasks are also time-consuming and/or boring for humans to perform. For instance, human visual inspection has been estimated to account for 10% or more of the total labor costs for manufactured products [109]. Human inspectors also often reject products based on the satisfaction of quotas rather than the actual defect le

for detection b

and improved

environments

Currently.

devices that r

several times

stoppage. Sin

also a lower l

Tradition

only the fina

certain advar

cause machi

they occur, r

that a fault f

25 cents to

components

involving m

involves sev

impossible [

presented b

Feasibility

Although a

reducing co

must run in

formally de

actual defect levels [45]. Furthermore, some defects in manufactured parts are too subtle for detection by the human eye [124]. In contrast, machine vision results in labor savings and improved quality. Finally, automated inspection allows objects to be inspected in environments unsafe for people.

Currently, many automated inspection tasks are performed using contact inspection devices that require the part to be stopped, carefully positioned, and then repositioned several times [98]. Machine vision can alleviate the need for precise positioning and line stoppage. Since machine vision inspection operations are, in general, non-contact, there is also a lower level risk of product damage during inspection [74].

Traditionally, most industrial inspection has focused on product inspection; usually, only the final assembly of the product is inspected. However, process inspection offers certain advantages. Without in-process inspection, for example, parts that fit poorly can cause machines to jam or break, interrupting assembly. If defects are not detected as they occur, material, time, energy, and labor are also wasted [159]. Zeuch has estimated that a fault found on bare printed circuit boards immediately after fabrication costs only 25 cents to repair, but if the same fault is found only after the board is fully loaded with components, then the repair cost increases to approximately $40 [194]. Assembly processes involving many parts pose a serious challenge to human inspectors. If the assembly process involves several operations per second, process inspection by humans is probably physically impossible [159]. One system that inspects the output of several assembly stages has been presented by Tsatsoulis and Fu [175]. Details of this system are given in Section 2.2.3.

Feasibility of automated inspection

Although automated inspection might seem to be a panacea for improving quality and reducing costs, it may not always be feasible. For automated inspection to be feasible, it must run in *real-time* and be consistent, reliable, robust, and cost-effective. It is difficult to formally define what is meant by real-time inspection, although Van Gool et al. [178] have

suggested
bottleneck
would req
inspection
tasks, such
take as lon
and time-co
developmen
illumination
achieving r
process to b
when the ap
very consist

The comp
placement of
belts, and ro
parts are plac
object silhoue
constraints ar
have been suc
assembled qui
constraints is n
to obtain well-
automated insp
procedure must

The flexibil
Bourne [187] ha

suggested a working definition "that the [visual inspection system] should not be the major bottleneck for reducing cycle time or robot operation speed." On many production lines, this would require the inspection of several parts per second [32, 178]. For most assembly line inspections, the upper time limit for inspection is probably about 1 second. A few inspection tasks, such as printed wiring board inspection, are not as time-critical, however, and can take as long as several minutes [186]. Automated inspection systems are also expensive and time-consuming to develop. Development costs can be as high as $100,000 [178]. The development cost usually cannot be amortized over many systems, either, because special illumination, image analysis, and part orientation restrictions are usually necessary steps in achieving robust system performance [132]. This makes it necessary for the development process to begin afresh for each application. Therefore, automated inspection is feasible when the application has large part volumes, demands very precise measurement, requires very consistent inspection, or is in a hazardous environment.

The complexity of automated inspection procedures can be reduced by requiring precise placement of the objects to be inspected. Positioning aids such as special fixtures, conveyor belts, and rotating tables have been used for this purpose. In some inspection systems, parts are placed on surfaces made of transparent glass or plastic to allow easy extraction of object silhouettes through backlighting [186]. Backlighting and other sensing environment constraints are appropriate for the inspection of simple objects. Constraints of these types have been successfully applied to the high-speed inspection of some products that are assembled quickly in large volumes. Unfortunately, using positioning aids and lighting constraints is not necessarily useful for more complex shapes; it is difficult and expensive to obtain well-jigged and uniformly lit parts on the factory floor [101]. Furthermore, for automated inspection to compete with the flexibility of human inspectors, the inspection procedure must allow the object to be in an arbitrary position and orientation.

The flexibility of human inspectors should not be underestimated. Wright and Bourne [187] have observed that machinists make many careful inspections and diffi-

cult accep
the part h
inch. For i
fixturing, a
are difficul
the machin
also contain
pounding th
these aircraf
and abilities
of the expert
challenging t

Logging De

One of the gre
logged for ea
production. A
in the producti
to be improved
integrated with
An integrated r
the future.

2.2 Litera

Many automated i
have surveyed som

cult acceptability judgments throughout the machining cycle. This is especially true when the part has a complex shape with tolerances on the order of several thousandths of an inch. For instance, aircraft parts have very complex profiles which require special tooling, fixturing, and numerical control programming. These parts are usually made of alloys that are difficult to machine. These alloys impose great stress on the machining tools, forcing the machinist to carefully monitor tool wear throughout the process. Many of the parts also contain internal cavities that reduce weight but also result in thin exterior walls, compounding the machining difficulty. Wright and Bourne claim that the machining process for these aircraft parts is "nondeterministic and unstable" and that the machinist's judgments and abilities are "irreplaceable secret ingredients of expert machining [187]." Automation of the expert inspective and adaptive capabilities of a human machinist would be quite a challenging task for a vision researcher.

Logging Defect Rates and Types

One of the great advantages of automated inspection is that defect rates can be automatically logged for each defect. This allows defect detection to be more closely connected with production. A high frequency of a certain type of defect might indicate that a tool or machine in the production process is malfunctioning, for example, or that the product design needs to be improved [74, 166]. Consumer complaints and field service records could also be integrated with the factory log of defects to monitor the complete lifespan of the product. An integrated record of product defects should enable better part design and production in the future.

2.2 Literature Review of Inspection

Many automated inspection systems have been presented in the literature. Chin and Harlow have surveyed some of the early work in automated visual inspection in [34]. Chin presented a second su

Comme
cluding che
plug dimen
valve body c
for impuriti
to check pac
objects for c
that can dete
developed [1
machine visi
applications.
the literature
have never be

2.2.1 Typi

A few of the
GEI, a machin
ductor packag
International h
automated insp
level and other
speed metal sur
None of the com
one cause for th

a second survey of papers published from 1981 to 1987 in [31].

Commercial inspection systems are available for a wide variety of inspection tasks, including checking engine blocks, tool wear, automobile wheel hub structural integrity, spark plug dimensions, dimensions of ceramic supports for CRTs, and automatic transmission valve body dimensions [117]. Automated inspection systems have also been used to search for impurities and fractures in casting processes, to inspect letters and numbers on labels, to check packaging and content of pharmaceutical and edible products, and to inspect glass objects for cracks and bubbles [20, 81, 141] or for presence of loose slivers [8]. A robot that can detect some of the defects in nuclear reactor vessel shells and nozzles has also been developed [18]. The majority of the automated inspection systems have been developed by machine vision companies while others have been developed by manufacturers for in-house applications. In this section, many of the inspection systems that have been presented in the literature are reviewed. However, most of the automated inspection systems probably have never been reported in the literature or have received only fleeting reference.

2.2.1 Typical Commercial Visual Inspection Systems

A few of the typical commercial visual inspection systems are presented in this section. GEI, a machine vision company, has developed the Gemini system for inspecting semiconductor packages for proper closure and correct placement of labels [141]. Machine Vision International has delivered systems for inspection of painted surfaces [8]. Sira has built automated inspection systems for monitoring photographic film [8] and for checking gloss level and other car body paint defects [178]. European Electronic Systems has built high-speed metal surface inspection systems for use in applications such as a rolling mill [8]. None of the commercial systems have been truly general-purpose, however, which may be one cause for the lack of business success for many machine vision companies.

2.2.2 Typical

Some large indus
example, built a s
has developed an
has developed an

2.2.3 A Taxo

This section pres
by the system.
binary images, g
modalities are al
in Table 2.1. A b
in Figure 2.3. (T
range data.)

Inspection Usi

Historically, mo
images [8, 32, 1
only simple coa
data are sufficie
their silhouettes
data [178]. It i
using inexpensi
lighting [186].
shadows, highli
orientation, text

2.2.2 Typical Industrial In-House Inspection Systems

Some large industrial companies have developed their own inspection systems. Saab, for example, built a system for automated inspection of flywheel castings [178]. Westinghouse has developed an automated visual inspection system for turbine blade inspection and Delco has developed an automated system for inspecting chip structure [117].

2.2.3 A Taxonomy of Inspection Systems

This section presents a taxonomy of inspection systems based upon the sensory data used by the system. We describe many of the systems presented in the literature that use binary images, gray level images, color images, and range images. Other possible sensing modalities are also discussed. A summary of some of the specific applications are exhibited in Table 2.1. A block diagram of one typical inspection system is shown later in the chapter in Figure 2.3. (The diagram, from [111], is of an automated inspection system that uses range data.)

Inspection Using Binary Images

Historically, most commercial vision systems for automated inspection have used binary images [8, 32, 178]. Many of these systems use images of limited resolution and perform only simple coarse verifications, such as testing for the presence of a part [94]. Binary data are sufficient to inspect many industrial objects which can be represented using only their silhouettes. Flat objects with no surface characteristics are also inspected using binary data [178]. It is economical to use binary images since these can usually be acquired using inexpensive sensors in conjunction with simple lighting arrangements, such as backlighting [186]. High contrast backgrounds and controlled lighting eliminate unwanted shadows, highlights, and noisy backgrounds but also make it difficult or impossible to use orientation, texture, and reflectance information in the inspection procedure [32]. The use

Input In
Binary
Intensity
Color
Range

Tab

[a] Used multiple bi

[b] Used multiple in

Input Image	Application	Date	Reference
Binary	Bolts	1992	[8]
	Color Picture Tube photomasks	1988	[84]
	Printed Wiring Boards (PWBs)	1983	[159]
		1985	[81]
		1986	[191]
		1989	[129][a]
	Screws	1982	[134]
	Shrimp Grading	1991	[96]
	Steering assemblies	1984	[175]
	Surgical staples and ligating clips	1990	[169]
	Tablets and capsules	1982	[48]
Intensity	Disk heads	1986	[139]
	Drill bit wear	1990	[107]
	Hot steel slabs	1983	[164]
	Integrated circuit chips	1985	[81]
	IC photomasks	1983	[186]
	PWBs	1984	[156]
		1985	[81]
		1986	[157]
		1986	[133]
		1989	[158]
		1990	[137][b]
	Reed switches and fuses	1982	[177]
	Semiconductor packages	1989	[141]
	Valve Spring key caps	1983	[150]
Color	Batteries	1989	[141]
	Citrus fruit grading	1989	[141]
	Denim dye streaks	1990	[41]
	Integrated circuit chips	1991	[188]
	Microelectric gate thickness	1987	[138]
	Poultry grading	1991	[42]
Range	Bolts, gear teeth, internal bores, engine valve dimensions and cylinder holes	1982	[142]
	Engine castings	1984	[153]
	Extruded aluminum profiles	1990	[40]
	Open-die forgings	1988	[187]
	PWBs	1988	[4]
	PWB solder joints	1988	[84]
	Ship Propellers	1984	[153]
	Solder joints	1988	[124]

Table 2.1. A Taxonomy of Selected Visual Inspection Systems.

[a] Used multiple binary images.

[b] Used multiple intensity images.

of binary images a
system's designers
sion systems use o
examination [8, 32

A number of b
developed an insp
on the photomask
using polygonal ap
of approximately f
manual inspection
profiles, and overa
systems that can b
presented a brief d
avoids special lig
The Itran 8000 in
comparison to a de
system which wa
Fuji Electric has
using features su
system was able t

Edward Weck
vision inspection
clips [169]. In t
of the inspection
extracted from ba
although each obj

Tsatsoulis and

of binary images also reduces the amount of data that needs to be processed, aiding the system's designers in meeting speed and cost requirements [32]. Typically, the binary vision systems use only simple verification schemes [81] such as pixel counting or boundary examination [8, 32] that can be executed very rapidly.

A number of binary vision systems have been presented in the literature. Hitachi has developed an inspection system that uses binary images to find ten different types of defects on the photomask for a color picture tube [84]. Olympieff et al. have inspected screws using polygonal approximations of object contours [134]. Their system operated at a speed of approximately five seconds per inspection and reduced consumer's risk from 0.5% for manual inspection to 0.05%. Diffracto has built a system that can inspect bolt threads, head profiles, and overall dimensions at a rate of three per second [8]. JAI of Denmark has built systems that can be used to inspect items such as rivets at high speeds [8]. Lapidus has presented a brief description of the Itran 8000, an automated visual inspection system which avoids special lighting and jigs by matching edge templates for object localization [101]. The Itran 8000 inspection system uses a number of pixels along an edge as the feature for comparison to a defect-free part. Bosch of Germany has also developed an image processing system which was used to inspect pin distances on electronic parts in binary imagery [173]. Fuji Electric has developed an inspection system that finds defects in tablets or capsules using features such as area and contour length extracted from binary images [48]. The system was able to inspect at a rate of five capsules per second.

Edward Weck, Inc., a subsidiary of Bristol-Meyers Squibb, is developing a machine vision inspection system for dimensional verification of surgical staples and ligating clips [169]. In this critical inspection task, 100% inspection is required. Few details of the inspection system are available, although it appears to use boundary information extracted from backlit objects. The objects have some freedom of position and orientation, although each object must be in approximately the same pose.

Tsatsoulis and Fu [175] have developed an in-process automated inspection system

that inspec
steps for T
two camer
real-time c
times rang
was allow
positions c
the assem
correctly.
presented

Kassl
packing c
approxim
other sta
or not.

PWB Ins

Due to i
wiring b
inspecti
is alread
human
inspecti
of the n

Ma
defects
flaws,

that inspects the dimensional tolerances and assembly integrity for 16 of the 24 assembly steps for TRW's Hydraulic Steering System. Their system collected intensity data from two cameras and thresholded the data to obtain binary images. This system ran in near real-time on a VAX 11/780, with an average time of 3 seconds per inspection step, with times ranging from 0.7 to 8.0 seconds per inspection. Some freedom in part positioning was allowed, but the system occasionally had difficulty inspecting parts in unexpected positions or orientations. The performance was quite good, however, in verifying most of the assembly steps; for twelve of the assembly steps, 100% of the defects were classified correctly. Unfortunately, little information about their defect detection algorithms was presented.

Kassler et al. [96] have developed a machine vision system for grading inspection and packing of shrimp for the Australian prawn industry. Their system operates at a rate of approximately 20 shrimp per second. It uses silhouette area to grade the shrimp and uses other statistics to determine the shrimp's orientation and to find if the shrimp is deheaded or not.

PWB Inspection

Due to increasing miniaturization and growing complexity, human inspection of printed wiring boards (PWBs) has become quite difficult and time consuming. In fact, the cost of inspection may equal the cost to fabricate the board [157]. Automated PWB inspection is already cost-effective because the inspection systems are faster and more accurate than human inspectors [141]. Soon, the complexity and miniaturization of PWBs may make inspection possible only through automated systems [4]. PWB inspection is considered one of the most important applications of machine vision.

Many algorithms and techniques for inspecting PWBs have been presented. Typical defects which must be detected include over-etchings (opens), under-etchings (shorts), flaws, cracks, holes, etc. Geometric features, such as conductor length and width, hole and pad dian

defects [186]

image subtra

Hunt has

Baird at Gen

cracked or fr

of IC corners

approach" fo

checked asse

ideal edges.

template mat

5 × 5 binary

appeared to c

Ye and D

spection. T

width. Their

cated boards

additional st

contained sm

Texas Ins

Measuremen

correctness c

used for insp

inspecting ar

minutes. The

of features su

geometric pa

and pad diameters, and distances between conductors, can be examined to find these defects [186]. Two well-known techniques that have been used to find the defects are image subtraction and feature-based approaches [186].

Hunt has presented an overview of several of the PWB/IC inspection systems in [81]. Baird at General Motors, for example, developed an IC chip inspection system that finds cracked or fractured chips by thresholding images and then matching against templates of IC corners. Hseih and Fu developed a top-down and model-driven "tree-like syntactic approach" for inspecting IC chips. Chin et al. developed a PWB inspection system that checked assembly correctness using graph matching of extracted edges against prestored ideal edges. Krakauer and Pavlidis developed a PWB inspection system that used binary template matching with a quick lookup table technique. Jarvis also inspected PWBs using 5 × 5 binary templates. Jarvis's system performed additional inspection in regions that appeared to contain a defect.

Ye and Danielsson [191] have also used binary images for printed wiring board inspection. They have developed two algorithms for checking conductor and insulator width. Their technique performed connectivity-preserving shrinking on images of fabricated boards and then compared the resultant images with those of an ideal board. Several additional steps allowed the techniques to avoid false rejection of boards whose images contained small protrusions or digitization noise.

Texas Instruments developed a PWB inspection system in the early 1980's called Parts Measurement System (PMS) [159]. This system was designed to inspect the dimensional correctness of parts before their assembly into printed circuit boards, but could also be used for inspecting molded, stamped, or die-cut metal and plastic parts. The features for inspecting an 8" × 8" printed circuit board could be programmed in approximately 30 minutes. The time per board inspection was about three minutes. PMS allowed inspection of features such as holes, slots, printed lines, conductors, etch patterns, corners, edges, and geometric patterns. The system was able to read numerical control (NC) programs written

in APT, an N
sequence to ve

Skaggs [1
compute lengt
measure and c
(height) of a p
PMS could in
reliability and
part throughpu

Ninomiya
defects in via
as 30 μm and
silhouette, or
against predete
shape perfectly
reported perfec
one occurrence

In general,
software requi
inspections of s

Inspection Us

Several systems
inspection. Ther
on intensity ima
conditions [32].
limited capabilit

in APT, an NC programming language, and could automatically produce an inspection sequence to verify the success of the NC operations.

Skaggs [159] reported that the PMS system required about 0.8 seconds to measure and compute length (with an accuracy of ±0.0002" at distances up to 10"), 1.8 seconds to measure and compute the diameter of a hole, and 3.6 seconds to measure the z component (height) of a point. Angles between lines could be measured with an accuracy of ±0.10°. PMS could inspect parts up to 45 × 36 × 12" in size. One application of PMS improved reliability and increased inspection rate by 40 to 60 times, resulting in a 40-fold increase in part throughput. That application also saved $120,000 annually in overhead.

Ninomiya et al. [129] have presented a PWB shape inspection system that detects defects in via hole fillings on IC chips. It detected concave and convex defects as small as 30 μm and lack-of-filling defects as small as 10 μm. To find defects in the chip, two silhouette, or "shadow," images were acquired from different viewpoints and compared against predetermined criteria. Their method was fast but it was unable to recover $2\frac{1}{2}$ D shape perfectly. Nevertheless, it is very suitable for inspection of simple objects. They reported perfect defect detection for 161 sample defects with a false alarm rate of less than one occurrence per IC sheet.

In general, binary inspection systems are advantageous because their hardware and software requirements are usually modest. Binary inspection is inadequate for many inspections of surface characteristics and 3D shapes, however.

Inspection Using Gray-level Images

Several systems have been presented in the literature that use gray level intensity data for inspection. There are few systems, however, that are capable of performing visual inspection on intensity images in complex industrial environments with dirt and unfavorable lighting conditions [32]. Most of the industrial inspection systems that use intensity imagery have limited capability and use image subtraction or localized histogramming methods for defect detection [8].

Vanderheydt et al. [177] developed a system in the early 1980's for inspection of reed switches, flash tubes, and fuses. Few details were presented about this system, although the authors have stated that graph matching of intensity and width profiles was used to identify defective objects.

Winkler has developed an automated system for inspection of photomasks used for fabricating integrated circuits [186]. The photomasks must be examined for defects such as registration errors, dimension variations, and other visual defects. Winkler's system was designed to replace a manual inspection process using a microscope that took approximately thirty minutes. Feature-based approaches using color separation were used in the automated system.

Ratcliff et al. [143] have developed an experimental system for defect detection on ground metal components. They detect defects first at a coarse resolution and then zoom in on any potential defective regions to find the size and location of the defect. Their system performed the inspection using images collected from several known viewpoints. The position of the part was also known *a priori*.

Rossol developed an intensity-based inspection system called KEYSIGHT to inspect valve spring assemblies on engine heads for the presence of valve spring cap keys [150]. The variation in the location of the assembly is greater than one-half of the key width, so KEYSIGHT first determined the assembly location and then inspected for the presence of the keys. The assembly's location was determined by thresholding and thinning an edge image. The object center was found using symmetry. The location was then verified using a template constructed from the edge image of an ideal valve spring cap assembly. The keys on the assembly were located by examining the intensity profile along a circle which passes through the keys. Since missing keys exhibited large dark areas in this intensity profile, high rates of dark pixels along the circle indicated missing keys. KEYSIGHT was not programmablc and was only applicable to the valve key inspection task [150]. The system

was robust with
keys, however.

Intensity im
wear is usually
that machinists
the tool edge is
remain usable [
a multifaceted
would use the
multifaceted, h
toolmaker's mi
areas of the dr
high wear usin
measurement
deviation from

Petkovic et
disk (magneti
a machine vis
surface. In ad
available. In
defect size an
generated fro
level features
dirt, contamin
database of th
system requir
and produce

was robust with respect to reflectance variations, different colors, and oil films covering the keys, however.

Intensity images have also been used in the inspection of tool wear. Currently, tool wear is usually monitored manually by a machine operator. Wright and Bourne have noted that machinists occasionally examine tool points during machining to make certain that the tool edge is still acceptable and to estimate the time duration in which the tool will remain usable [187]. Liu [107] has developed a system which examines the flank wear of a multifaceted drill bit to determine if the tool is still usable. A typical test for drill wear would use the width of the drill flank wear, but because the drill in Liu's application was multifaceted, he used drill wear area instead. Drill wear area can be measured through a toolmaker's microscope, but that is very tedious and time-consuming. Liu found that worn areas of the drill had a higher reflectivity than unworn surfaces, so he located regions of high wear using an intensity histogram. Liu compared his system's performance to manual measurement of areas under a microscope and found that his system exhibited a 3.5-4.5% deviation from the performance of a human expert.

Petkovic et al. [139] have developed an automated visual inspection system for detecting disk (magnetic storage media) head defects. Disk head inspection is quite challenging for a machine vision system because it is necessary to rapidly locate small defects on a large surface. In addition, there are usually only a few training samples of defective disk heads available. In Petkovic et al.'s system, images were gathered through microscopes to find defect size and position. Defects were classified using boundary fitting and a set of rules generated from a model of a defect-free disk head. Object size, shape, position, and grey-level features were used to determine the most likely labeling for the defect (e.g., noise, dirt, contamination, voids, etc.). The list of possible defects was then matched into a rule database of the disk head's engineering specifications to produce a list of violations. Their system required 3.5 to 7 minutes per disk head to acquire the image, perform all processing, and produce the list of violations. Petkovic et al. tuned and tested the system on over

10,000 imag

Several sy
gray-level im
for inspection
microelectron
that are depen
intensity imag

Odawara e
incorrect line
the size and p
that their syste
performed at v

Silvén et al.
tion system for
somewhat robu
by picking seve
image generated
approach for de
controlled wirin
patterns. In the
a structural grap
occurring defect
a high matching

Park and Tou
printed circuit bo
512 x 512 inten
Features were ext

10,000 images and reported 99% accuracy for defect classification.

Several systems have been presented in the literature for defect detection in PWBs using gray-level images. Sreenivasan et al. [160], for example, have recently presented a method for inspection of the ball and wedge bonds that connect bond pads to the lead fingers of microelectronic circuit chips. Their method uses histogramming with varying thresholds that are dependent upon the region (e.g., bond region, background, and pad region) of the intensity image. Bond size and shape are inspected after segmentation.

Odawara et al. [133] have used feature extraction to detect micro defects, including incorrect line width, nicks, and pinholes. Dimensional errors of larger patterns, such as the size and position of traces, were detected using design rule checking. They claimed that their system operated at video rate, although they did not state which functions were performed at video rate nor did they indicate the accuracy of their system.

Silvén et al. [156, 157, 158] have developed an experimental CAD-based visual inspection system for verifying printed circuit boards using edge information. Their system is somewhat robust with respect to board alignment. Minor misorientations were resolved by picking several line segments and rotating the board to align the sensed image with an image generated from the model [156]. The system used both a statistical and a structural approach for defect recognition. The statistical analysis performed well for simple, tightly controlled wiring patterns, while the structural analysis worked better in complex wiring patterns. In the structural approach, defects were recognized by a depth-first matching of a structural graph derived from a sensed image with stored structural graphs of commonly occurring defects. Heuristic rules, such as commencing the search from nodes which have a high matching score, were used to simplify the matching [157].

Park and Tou [137] have presented a CAD-based inspection method for examining printed circuit board solder joints. Fifteen features were extracted from four frames of 512 × 512 intensity images. Each frame was collected using a different light source. Features were extracted using CAD models that contained the sizes and locations of ideal

solder joints and
categorize defects
data of defective
a false alarm rate
system used a sin
per frame. Becau
fast.

Suresh et al. [
measures surface
cast steel requir
steel is defect-fr
is conditioned to
inspection is us
thus resulting in
to extract edge f
delivered a peak
Roberts edge op
connected edges
at a rate of 4 in
set of labels and
minicomputer d
imperfections.
decision tree. U
it was necessary
task in the haza

Applied Int
cent lamp inspe

solder joints and pads. Park and Tou used a fuzzy set-based classification algorithm to categorize defects into one of seven classes. The classifier was designed using training data of defective joints. They reported an average correct classification rate of 93.9% and a false alarm rate of 5.6%. In 3.8% of the cases, the defect could not be identified. Their system used a simple lighting arrangement and was able to process up to 250 solder joints per frame. Because four images were used, however, the inspection process was not very fast.

Suresh et al. [164] of Honeywell have developed a system which finds, identifies, and measures surface flaws on hot steel slabs in a steel mill. Manual inspection of continuously cast steel requires the steel to be cooled after it has been cast. After inspection, if the steel is defect-free, it is reheated for further processing. If the steel contains defects, it is conditioned to remove the defect and then reheated for more processing. Automatic inspection is useful because it allows the slab to be inspected with little or no cooling, thus resulting in large energy savings. The Honeywell system used a four-processor array to extract edge features from the slab as it traveled underneath a sensor. One processor delivered a peak performance of 10 MFLOPs and performed the edge detection using the Roberts edge operator. The other three processors, which operated at 1 MFLOP each, found connected edges and performed object labeling and feature extraction. The slab traveled at a rate of 4 inches per second, requiring a processing rate of 546 Kpixels/sec. As each set of labels and extracted features were generated, they were downloaded to a Honeywell minicomputer delivering 1 MFLOP peak performance that identified and classified the imperfections. Defect classification was performed using syntactic rules and a binary decision tree. Unfortunately, error rates of the system were not presented. Before each use, it was necessary to fine-tune system performance using training data, which is not a trivial task in the hazardous environment of a steel mill.

Applied Intelligence Systems and AECL Advanced Systems have developed a fluorescent lamp inspection system for General Electric [5]. This system uses 6 cameras, controlled

lighting, and thre
forms 100% insp
for lamp pose, al
of the lamp. Tw
All dimensional

West, Fernan
for flat object ins
planar and cylind
pose clustering
circle centers, an

Park and M
inspection of ob
tolerances and f
vertices. Their
on a CAD mod
Horaud's [17] lo
and morphologi
in a hypothesize
and Mitchell res
degrees of freed
extracted from
and Mitchell dic

Inspection Usi

As Daley and R
and food inspect
color inspectors a

lighting, and three SIMD array computers to inspect for nine defects in lamp bases. It performs 100% inspection at a rate of 8000 lamps per hour. The system is somewhat flexible for lamp pose, although the assembly line imposes some constraint on the general position of the lamp. Two salient features on the ends of the lamp are used to determine the pose. All dimensional inspections on the lamp are made with respect to these datum features.

West, Fernando, and Dew [184] have developed an experimental CAD-based system for flat object inspection in intensity images. Their system inspects positional tolerances of planar and cylindrical surfaces. They determine approximate object pose using Stockman's pose clustering technique [162] and then inspect distances between edges, positions of circle centers, and circularity.

Park and Mitchell [136] have developed an experimental system for feature-based inspection of objects in intensity images. Their system is designed to inspect dimensional tolerances and for the presence or absence of features such as holes, slots, edges, and vertices. Their research is aimed at automatically generating an inspection plan based on a CAD model of a part. They estimate part position using a variant on Bolles and Horaud's [17] local feature focus method. Specifically, they use holes/circles, line fitting, and morphological image operations as well as the pairwise relations between these features in a hypothesize-and-verify scheme to recognize the part and determine its position. Park and Mitchell restrict object position and orientation, however; i.e., they do not allow six degrees of freedom in pose. After the part has been identified, the dimensional tolerances extracted from the CAD model by the inspection planning algorithm are examined. Park and Mitchell did not quantify the performance of their technique.

Inspection Using Color Images

As Daley and Rao [41] have observed, color is an especially important factor in apparel and food inspection where changes in shade or variations in color indicate a defect. Human color inspectors are not reliable, primarily because humans do not have a very good memory fo

A f

dev

the r

as li

lens a

to the

So

inspect

bruises

machine

injury. T

Poole

of the lab

tips were c

the system

proceeded t

soon as the

Daley ar

and discolor

are pinkish o

on denim by

then segmentin

and Carey [42]

color images of

T800 transputers

examine one bird

for color. Therefore, color inspection tasks are very appropriate for automated systems. A few experimental automated inspection systems using color images are currently under development. One barrier to more widespread application of color inspection systems is the necessity for more computing power, intricate optics design, and lighting. Factors such as light source intensity, color temperature, angle of illumination, and magnification and lens aperture are crucial for accurate color monitoring [138] due to the increased sensitivity to the energy distribution from the light source [41].

Some of the current commercial agricultural applications of automated color vision inspection include systems that check the color of bell peppers or examine apples for bruises [141]. Sunkist Corporation has developed a citrus grading machine that uses machine vision to grade oranges and lemons according to size, color, blemishes, and frost injury. This machine grades eight pieces of fruit per second [141].

Poole [141] has reported an automated inspection system for batteries. Color registration of the label, battery length and width, and correct formation of the positive and negative tips were checked by this system. Since 80% of the defects were color registration errors, the system first checked for that defect, and, if no registration errors were encountered, proceeded to search for other defects. The battery was removed from the assembly line as soon as the first defect was detected.

Daley and Rao [41] are exploring automated inspection to find dye streaks on denim and discolorations on poultry skin. (Specifically, they wish to detect overscalds, which are pinkish or reddish skin in the birds' neck area.) They were able to find dye streaks on denim by transforming color data into HSI (hue, saturation, and intensity) space and then segmenting the image using a histogram-based thresholding on the hue values. Daley and Carey [42] have also used thresholding and morphological operations to segment HSI color images of poultry. Their current system achieved near real-time performance using 4 T800 transputers. Their goal is to achieve real-time processing (human inspectors currently examine one bird every two seconds) using an array of 16 T9000 transputers.

Parthasarath
vision for rapid i
usually fabricate
produce colors
two schemes to
resolution of 20
RGB color space
optimistic becaus
of the training sa

Xie and Beni
of IC chips. Part
region around the
concentric rings c
of the defect by c

Inspection Usin

For many applicat
Pryor has noted t
intensity (binary,
negligible informa
in reflected light a
listed gears, femal
information (depth
of parts that have
measurement of d
it explicitly repres
explicit surface in

Parthasarathy et al. [138] have developed an experimental inspection system using color vision for rapid inspection of microelectric gate oxide thickness. Microelectric structures are usually fabricated using a thin film that is optically transparent. Optical interference effects produce colors that are characteristic of film thickness. Parthasarathy et al. presented two schemes to determine the film thickness. One method determined thickness with a resolution of 20 angstroms in less than 100 msec using a nearest-neighbor matching in the RGB color space. It achieved an accuracy of approximately 90%, but this percentage was optimistic because every test sample that was examined was of a thickness identical to one of the training samples.

Xie and Beni [188] have used fuzzy clustering for detecting defects in the thin film layer of IC chips. Particles on IC wafers cause the film thickness to vary nonuniformly in the region around the particle, which produces light interference patterns that appear as several concentric rings of different colors around the defect. They were able to estimate the size of the defect by counting the number of color rings in the interference pattern.

Inspection Using Range Images

For many applications, inspection using binary, grey-level, or color images is impractical. Pryor has noted that surface inspection of turbine blades is usually not possible using intensity (binary, grey-scale, or color) imagery because silhouette information provides negligible information about the blade surface and intensity data can contain variations in reflected light as large as 100,000:1 over the surface of the blade [142]. Pryor has listed gears, female thread forms, and bore inner diameters as other surfaces for which 3D information (depth or range images) is necessary for inspection. Dimensional inspection of parts that have etched height variances, warp, or reliefs also requires accurate 3D measurement of depth [159]. The greatest advantage in using range data, though, is that it explicitly represents surface information [92, 111]. As Marshall has observed, when explicit surface information is available, the shape information can be used for efficient

matching. In c
inspection syst
acquisition are
there do not app
that have been

Robotic Vis
processing syst
Their claim of
was reported t
800,000 points
surfaces for the
1250 dimension
inexact workpie
Caterpillar.

An automat
die forgings [18
a rotary turntab
measurements a
by the system.
critical locations

Several PWE
the literature. H
on printed circui
points per second
a system that use
presence or absen
triangulation arra

matching. In contrast, matching on edges, which is the practice of many of the grey-level inspection systems, discards much of the surface information [111]. Range images and their acquisition are discussed in more depth in Chapter 4 of this dissertation. Unfortunately, there do not appear to be any complete 3D industrial inspection systems for complex objects that have been placed on-line yet [32].

Robotic Vision Systems developed a structured light sensor and MC 68000-based image processing system in 1984 which were claimed to acquire 3D data in real-time [153, 154]. Their claim of real-time operation is somewhat dubious, however, because the sensor was reported to take 40 minutes to scan all six sides of a cubic object (although over 800,000 points were scanned). The sensor has been used for inspecting large propeller surfaces for the U.S Navy; engine castings for Cummins Engine (in this application, over 1250 dimensional inspections were required); and for adaptive weld seam following (with inexact workpiece positioning) on production lines at General Motors, Westinghouse, and Caterpillar.

An automated 3D inspection system has also been developed for inspection of open-die forgings [187]. The 3D image was constructed by backlighting the object and using a rotary turntable to collect silhouette data from 360 viewpoints. Three cross-sectional measurements and two vertical measurements were extracted from the image and verified by the system. The system replaced a person who had manually measured some of the critical locations on the forging using calipers.

Several PWB or IC inspection systems that utilize range data have been reported in the literature. Hitachi, for example, has developed a system that examines solder joints on printed circuit boards [84]. Their system detects four types of defects and inspects ten points per second using a 3D structured light arrangement. Badami et al. [4] have presented a system that uses dense range data to inspect leads on a PWB. The system checks for the presence or absence of leads and verifies lead height and angle when necessary. They used a triangulation arrangement for data acquisition with a depth resolution of 0.0006 inches that

measured 120,0
details were pres

Dahle et al.
using a structure
generated model.
Their goal is an
rate is not yet th

Pryor has dev
are discussed in [
achieve high accu
in gears at a rate
and sorted bolts
system inspected
inspection for sur
than 6 seconds. (i
rate of 4200 valv
assemblies were
microcomputers.
80 image sensors
blockage. A part

Many systems
images are well-su
indicators are com
system that used st
number of point li

*Pitch line runout r

measured 120,000 points per second (i.e., about 0.5 in^2 of the board per second). Few other details were presented about how the system functioned or how accurately it performed.

Dahle et al. [40] are developing a system that inspects extruded aluminum profiles using a structured light arrangement. Their system uses information extracted from CAD-generated models for inspecting surface planarity, gap dimensions, and angular deviations. Their goal is an inspection rate of 1-2 objects per second, although the current inspection rate is not yet that fast.

Pryor has developed several inspection systems using structured light, some of which are discussed in [142]. These systems were developed in the early 1980's but were able to achieve high accuracy. (i) One system inspected pitch line runout *and tooth spacing errors in gears at a rate of one pair of mating gears every 10 seconds. (ii) Another system qualified and sorted bolts on ten different dimensions at rates of 10,000 bolts per hour. (iii) A third system inspected the internal bores of a power steering housing, performing a complete inspection for surface defects such as pits, scratches, voids, tool marks, and cracks in less than 6 seconds. (iv) 12 cameras were used to inspect the dimensions of engine valves at a rate of 4200 valve inspections per hour on the production line. (v) Power steering worm assemblies were inspected at a rate of 1500 parts per hour using two cameras and two microcomputers. (vi) Threaded and machined holes in cylinder heads were inspected using 80 image sensors to check for the presence of each thread in every hole and to detect hole blockage. A part throughput rate of 600 parts per hour was achieved.

Many systems have also been created for automatic inspection of solder joints. Range images are well-suited for solder joint inspection because the joints and their key process indicators are complex geometric shapes [147]. Nayar et al. [124] recently developed a system that used structured highlighting, which consists of illuminating the object by a large number of point light sources, to compute local surface orientations. These orientations

*Pitch line runout refers to the eccentricity in the line of contact between gear teeth.

were utiliz

The solder

required 12

all computa

a rate whicl

General-Pur

All of these

in a single ap

laboratories a

system for gro

and feature rel

multiple views

faces, althoug

match the mode

Marshall ma

depth-first searc

rigidity constrair

three or four) sur

model and scene

than the first few l

which was then re

been used in severa

including those of

[1]EGIs are discusse

[2]The interpretation

[3]The local-feature

Bolles and Horaud [1

terminology.

were utilized to build an Extended Gaussian Image (EGI) representation[†]of the joints. The solder joints were then classified using features extracted from the EGI. The system required 12 seconds to inspect each joint. This inspection time included image capture and all computation steps. A misclassification rate of three percent of the joints was observed, a rate which is within the variance of human experts in solder inspection.

General-Purpose Inspection

All of these range-based inspection systems are highly optimized and constrained for use in a single application. Several experimental systems are under development at research laboratories and universities that are more general, however. Marshall [111] has developed a system for gross feature inspection, including verification of hole presence, part dimensions, and feature relationships (e.g., distances and angles) using dense range data collected from multiple views. His system inspected objects composed of planar, cylindrical, or spherical faces, although only planar surfaces (which can be extracted more reliably) were used to match the model to the sensed object.

Marshall matched the surfaces in one of the sensed views to the model surfaces using a depth-first search of an interpretation tree[‡] The tree was pruned to a manageable size using rigidity constraints; i.e., the model-to-view transformation generated after several (usually, three or four) surfaces had been matched. Specifically, a "current transformation" between model and scene was computed afer several surfaces had been matched. Searches deeper than the first few levels were pursued only at nodes that satisfied the current transformation, which was then refined using the most recent matching. Rigidity constraints have also been used in several of the approaches to 3-D object recognition presented in the literature, including those of Faugeras and Hebert [49] and Chen and Huang [27][§] The best match

[†] EGIs are discussed in more detail in Chapter 6 of this dissertation.

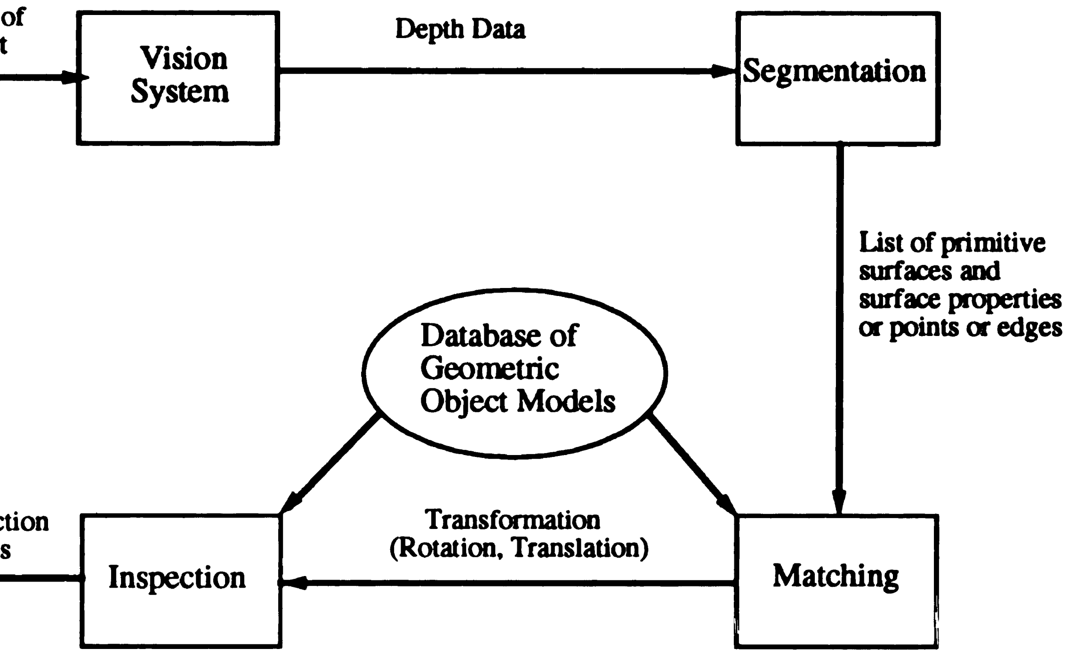
[‡] The interpretation tree search technique is discussed in detail in Chapter 6.

[§] The local-feature-focus method of Bolles and Cain [16] and the 3DPO object recognition scheme of Bolles and Horaud [17] also employ rigidity constraints, although Bolles, Cain, and Horaud prefer different terminology.

Figur

in a single vi
unnecessary t
other views.

The other
spected for err
execution time
his goal. Marsl
if a model of t
that are visible
inspected only
detected were a
may be limited.
general approac


of
t
Vision System
Depth Data
Segmentation
List of primitive surfaces and surface properties or points or edges
Database of Geometric Object Models
ction
s
Inspection
Transformation (Rotation, Translation)
Matching

gure 2.3. Schematic diagram of Marshall's [111] 3D Inspection System.

view provided the position and orientation of the object. Therefore, it was y to perform matching between the model and the sensed object surfaces in the s.

her views were used to check geometric tolerances. Specifically, Marshall in- errors in position, size, and shape of model primitives. No details about system imes were presented, although real-time performance did not seem to have been larshall's work is significant because of its generality (any object can be inspected of the object exists, provided that the object contains several planar surfaces sible to the sensor) and its use of multiple range images. Marshall apparently only two fairly simple objects, however. Also, many of the defects his system ere artificially induced on the objects, so the general application of his method iited. A diagram of Marshall's inspection method is shown in Figure 2.3. The proach outlined in this diagram is similar to that followed in our work.

Inspection Usi

Other sensing m

is often used to i

technique based

gren [72] has dev

drill holes in met

X-ray and ultraso

the Canny edge d

Alternate Modalit

Chan et al. [25].

automated visual

in food, for instan

the food sample's

foods. Other adv

nature of the ima

considerations, ho

Chan et al. [25

(NMR) imaging fo

very expensive, ho

Gamma rays ar

et al. [25] have not

present several dis

of lower contrast t

X-rays, there are d

machines cannot b

Inspection Using Other Sensing Modalities

Other sensing modalities are also possible for inspection. For example, X-ray inspection is often used to inspect complex industrial parts [72]. Noble et al. [132] have proposed a technique based on deformable templates to inspect drill holes in X-ray images. Hedengren [72] has developed real-time algorithms to verify the presence, size, and location of drill holes in metal using X-ray images. Jain and Dubuisson [87] have located defects in X-ray and ultrasonic C-scan images of composite materials using adaptive thresholding and the Canny edge detector.

Alternate Modalities for Food Inspection

Chan et al. [25], have examined the suitability of other sensing modalities for use in automated visual inspection of foodstuffs. X-rays could be used to detect foreign bodies in food, for instance, although only foreign bodies whose density is higher or lower than the food sample's density can be detected. X-rays are also useful for texture analysis of foods. Other advantages in using X-ray images for inspection include the high contrast nature of the images and the potential for real-time imaging. There are cost and safety considerations, however, in the use of X-rays.

Chan et al. [25], also discuss the potential application of nuclear magnetic resonance (NMR) imaging for inspection of the texture and composition of food. NMR equipment is very expensive, however.

Gamma rays are similar to X-rays but can be used to inspect denser materials. Chan et al. [25] have noted that while gamma ray images can be acquired in real time, they also present several disadvantages. One disadvantage is that gamma ray images typically are of lower contrast than X-ray images. There are also several safety considerations. Like X-rays, there are dangers from gamma radiation, but unlike X-ray machines, gamma ray machines cannot be switched on and off. This makes transportation and storage difficult and also presc

Ultrasoun
foods with no
ultrasound alth
special transdu

Another po
(infrared) imag
as five images
considerations
limited sensitiv

Ultraviolet
ultraviolet ligh
exhibit changes
yellow fluoresc
color. Ultraviol
can cause eye d
automated visio
complexity to th

Defect Enhancem

Nondestructive v
penetrant testing
cast or forged pa
defect-free regio
dye to make crac
sprayed on a surfa
that colors the ar

and also presents constant safety concerns.

Ultrasound images may also be useful for foodstuff inspection, particularly for packaged foods with no air content and high water content [25]. There are no safety concerns about ultrasound although it is not well-suited to many industrial environments because it requires special transducers for imaging.

Another potential sensory source for automated visual inspection of foodstuffs is thermal (infrared) imaging [25]. Infrared (IR) sensors are flexible and easily recalibrated. As many as five images per second can be collected with current IR sensors. There are also no safety considerations about IR sensors. A disadvantage of IR imagery is that the sensors have limited sensitivity and produce images which usually require pre-processing.

Ultraviolet (UV) light can also be used for fluorescent analysis of certain foods. While ultraviolet light does not penetrate materials, some foods, such as eggs, milk, and fish, exhibit changes in fluorescence with freshness [25]. Fresh milk, for example, exhibits a yellow fluorescence while sour milk has a fluorescence which is white to grey violet in color. Ultraviolet images can be acquired in real-time, although high intensity UV light can cause eye damage or skin cancer. It also may affect some foodstuffs. Furthermore, automated vision systems using ultraviolet images require color processing, which adds complexity to the system software and hardware.

Defect Enhancement - Magnetic Particle and Liquid Penetrant Testing

Nondestructive visual inspection has also been conducted using magnetic particle and liquid penetrant testing. These two testing techniques are especially applicable to inspection of cast or forged parts [62, 97]. Both techniques increase the contrast between defective and defect-free regions of a part. Liquid penetrant testing (LPT) consists of applying a visible dye to make cracks, pores, and other surface defects more visible. Usually, a penetrant is sprayed on a surface, and after the penetrant dries, the surface is treated with a developer that colors the area of inspection white while also removing the penetrant from defects.

As a result,
penetrant dy
bright and s

Magneti
techniques [
particles to t
The part is th
particles are
defect locati
in castings ca
further enhan
other inspect
material stru

LPT and
detection of
thermal loads
the ability of

Gages and
by many com
Automated in
and Ferrano [
applications. T
light sources t
housings, turbi

As a result, defects usually appear as dark or red lines or spots [97]. Often a fluorescent penetrant dye is used which, when viewed under ultra-violet light, allows defects to appear bright and sharp [97].

Magnetic particle testing (MPT) is one of the most widely used contrast enhancement techniques [62, 97]. Magnetic particle testing consists of applying fine ferromagnetic particles to the object surface by dipping the object in a water- or oil-based suspension [97]. The part is then magnetized using either a longitudinal or circular magnetic field. Magnetic particles are attracted to regions where the magnetic field crosses a crack or pore, indicating defect location to a human operator. In a similar fashion, near-surface nonmetallic inclusions in castings can also be detected [62]. The object is often inspected under ultraviolet light to further enhance the contrast of the defects. Magnetic particle testing is preferable to many other inspection techniques because it is affected less by surface geometry, roughness, and material structure [62].

LPT and MPT testing are particularly important due to their ability to greatly enhance detection of surface defects. As Goebbels and Ferrano [62] have noted, mechanical and thermal loads as well as corrosion are directed at surfaces. Surface defects greatly reduce the ability of objects to withstand such loads.

Gages and machines to aid human inspectors in LPT or MPT have been developed by many companies. Magnaflux, for example, dominates the market for MPT machines. Automated inspection using LPT or MPT techniques has also been conducted. Goebbels and Ferrano [62], for instance, have developed automated inspection systems for many applications. Their systems use from one to four cameras under both visual and ultraviolet light sources to detect flaws in many small parts, including casting rods, coil springs, axle housings, turbine blades, and wheel hubs.

2.2.4 Inspe

Automated visu
first approach i
image) to the s
match the mod
object and com
model. If all o
approaches are

Template Ma

Automated insp
using template
value of pixels
free model. Th
synthetically g

One area in
visual inspecti
matching syste
(Their system
plate matching
The template n
photomasks wi
frequently occu
template match
that arise from
also been used

2.2.4 Inspection Schemes

Automated visual inspection techniques can be separated into two general approaches. The first approach involves matching a template of a defect-free model (often this is a synthetic image) to the sensed image of the object to be inspected. Objects which are defect-free match the model well. The second approach involves extracting features from the sensed object and comparing those features to a description or list of rules that describes an ideal model. If all of the rules are satisfied, the object is considered to be defect-free. The two approaches are described in detail in the following two subsections.

Template Matching

Automated inspection has been performed on object silhouettes and also on grayscale images using template matching schemes. These pixel-by-pixel matching schemes compare the value of pixels in an image of the sensed object against pixels in stored images of a defect-free model. The image of the defect-free model may be a sensed image or, alternately, a synthetically generated image, perhaps from a CAD model.

One area in which template matching techniques have been successful is the automated visual inspection of printed circuit boards [132, 158]. One example of a pixel-by-pixel matching system for printed wiring board inspection was proposed by Silvén et al. [158]. (Their system was discussed in more detail in Section 2.2.3 of this dissertation.) Template matching has also been used for inspection of integrated circuit photomasks [158]. The template matching was performed in that system by comparing images of fabricated photomasks with images of perfect photomasks. The system experienced difficulty with frequently occurring variations of patterns, however. This is a common complication in template matching approaches; there are natural (non-defective) variations between parts that arise from most manufacturing processes [132]. Template matching techniques have also been used for seam detection in industrial seam tracking applications [178] and for inspection of d

Pixel-level
detect many fa
portion of the
template matc
match the hunc
Schemes for r
learning algori
Template matc
the object is ur
practice, many
the approximat
often take a va
the template [1
example [147].
features suffer
the imagery.

Rule-based M

Most of the ind
rule-based comp
according to cla
rules, which, in r
perimeter, ratio
bounding box ar
schemes over ten
of templates [14

inspection of drill holes [132].

Pixel-level comparison schemes are flexible and are probably the only reliable way to detect many faults [156]. However, comparing low-level pixel values utilizes only a small portion of the potential information which could be used in inspection [156]. Exhaustive template matching is also quite time consuming; no matching strategy can efficiently match the hundreds of thousands of pixels commonly encountered in an image [111, 178]. Schemes for reducing the size of the template have been proposed, though, such as the learning algorithm based on statistical pattern recognition proposed by Lin et al. [106]. Template matching becomes painstakingly slow if the pose (orientation and position) of the object is unknown, or if the scale of the pattern to be matched is unknown [178]. (In practice, many template matching schemes are not very flexible because they assume that the approximate pose of the object is known.) Furthermore, "good" and "bad" parts can often take a variety of shapes, making it difficult to detect which objects deviate from the template [147]. This is one of the key problems in inspection of solder joints, for example [147]. On the other hand, most inspection techniques that rely on high-level features suffer from the difficulty and time-consuming nature of extracting features from the imagery.

Rule-based Methods

Most of the industrial inspection systems that have been reported in the literature utilize rule-based comparison schemes. Rule-based schemes make decisions about object quality according to classification rules [178]. The sensed object is checked against a list of design rules, which, in most industrial inspection systems, consists of features such as surface area, perimeter, ratio of perimeter to area, number of holes, area of holes, minimum enclosing bounding box area, maximum radius, minimum radius, etc. [178]. One advantage of these schemes over template matching is that it is unnecessary to maintain an extensive database of templates [147]. The rule-based methods suffer from the disadvantage that they are less adaptable t

rule-based schen

A Hybrid Met

Hybrid techniqu

ble. Noble et al.

and rule-based t

to examine drill

first determines

consisting of cur

template are loca

The model temp

evaluated to dete

essary to align th

ranges specified

collecting statisti

2.3 Summ

In this chapter, a

automated visual

less adaptable to design changes than pixel-level comparison schemes, however. Some rule-based schemes also require complicated schemes to eliminate false alarms [156].

A Hybrid Method

Hybrid techniques that combine template-matching and rule-based reasoning are also possible. Noble et al. [132], for example, have presented a variation on the usual template-based and rule-based techniques. Their technique uses deformable "templates" in X-ray images to examine drill holes. Strict pixel-based matching is not performed, rather their method first determines a coarse match between edges extracted from the image and a template consisting of curve primitives. After a coarse fit has been made, the curves in the model template are locally deformed to more closely match the edges extracted from the image. The model template is augmented with higher-level geometric constraint rules which are evaluated to determine acceptability. The part is accepted only if the deformations necessary to align the model template to the edges extracted from the image fall within the ranges specified by the constraint rules. Noble et al. established their constraint rules by collecting statistics on a large sample of defect-free parts.

2.3 Summary

In this chapter, a taxonomy of the stages of inspection was presented. An overview of the automated visual inspection systems reported in the literature was also presented.

CHA

CAD

The repres

for compu

tion. Each

object repr

that variou

application

allows for e

This ch

tations for

discussed.

3.1 Inc

Representati

stress and he

manufacturin

rendering [15

Many mar

CHAPTER 3

CAD Models

The representation of the three-dimensional, or volume, data from solids is very important for computer graphics, robotics, computer-aided design, and object recognition and inspection. Each of these application areas has different, sometimes conflicting, requirements for object representation. For graphical display purposes, objects need to be represented such that various renderings of the objects can be quickly generated, while for computer vision applications, it may be more important to represent a model of an object in a manner that allows for efficient matching of image data to a real-world model.

This chapter presents an overview of some of the requirements and popular representations for object recognition and inspection. Our approach to object modeling is also discussed.

3.1 Industrial Uses and Advantages of CAD Models

Representations for computer-aided design need to support many operations including stress and heat transfer study, volume and mass property calculations, computer-controlled manufacturing, interference detection, modification of design parameters, and fast graphics rendering [151, 116].

Many manufactured parts and assemblies are currently designed using computer-aided

design too
defined m
of an obje
encoding c
for produc
bled produ
augmented
desired qua
plicable for
that new obj

3.1.1 Tol

With the exce
pabilities for
tolerances, ho
there are often
is treated as a
tolerances are

Manufactu
current solid m
and CAD litera
inspection usua
CAD packages
based representa
traditional CAD
aspects of CAD
includes "form fea

design tools. The CAD models representing the parts and assemblies provide a ready, well-defined model for inspection [81]. Most CAD models provide a mathematical description of an object's shape, including an explicit parametrization of surface shape and an explicit encoding of inter-surface relationships. The CAD model's abstract representation is useful for product design and potentially useful for inspection operations that compare an assembled product to the ideal model designed by the CAD system. CAD databases can also be augmented with manufacturing information, including material type, geometric tolerances, desired quality of surface finish, and finish color, thus providing a unified description applicable for inspection. Another advantage of using CAD representations for inspection is that new object models can be quickly added to the current inspection framework.

3.1.1 Tolerances and Features

With the exception of a few recent releases, CAD modeling packages generally lack capabilities for representing and manipulating variational information such as dimensional tolerances, however [136, 111]. One problem with current dimensional tolerancing is that there are often ambiguities in interpretation depending on which area or point on the object is treated as a datum (or landmark) [111]. Datums (reference landmarks) and dimensional tolerances are discussed in more detail in Chapter 9.

Manufacturing features such as holes and slots usually are not explicitly represented in current solid models. (The notion of a "feature" is defined only vaguely in the manufacturing and CAD literature, usually as a region of interest on a part [116].) As a result, features for inspection usually have to be extracted from the CAD database. A few of the commercial CAD packages are beginning to support feature-based representations, however. Feature-based representations allow a higher level of abstraction than the geometric primitives of traditional CAD systems and form a connection across design, analysis, and manufacturing aspects of CAD [116]. Merat et al. [114] have identified a dichotomy of features that includes "form features" and "geometry-modifying features." The first class includes slots, through-hole

features that

The CAD

that supports

observed feat

in the CAD d

single 2D ima

Figure 3.1

block's surfac

1, 2, and 3 ar

surfaces 1 and

also defective i

to the "floor"

explicit represe

of the block ob

of primitives* w

operations coul

3.1.2 Defec

CAD models ar

It is likely that s

modeled geomet

attempt to determ

of common defec

be quickly introdu

*Primitive instanc

through-holes, ribs, pockets, etc. The other class includes chamfers, fillets, and other features that modify the geometry that the form features define.

The CAD modeling system being used in this research, IDEAS VI, is one package that supports feature-based representations of objects. Using feature-based models allows observed features extracted by a vision system to be directly compared against those stored in the CAD database. Park and Mitchell [136] have inspected simple flat 3D objects in a single 2D image using a feature-based CAD modeler.

Figure 3.1 is an example of a defect-free block which contains a slot. Three of the block's surfaces are labeled "1," "2," and "3." Surfaces 1 and 2 are coplanar while surfaces 1, 2, and 3 are parallel. Figure 3.2 is a defective instance of the block, however, where surfaces 1 and 3 are parallel but surface 2 is not parallel to surfaces 1 or 3. The slot is also defective in the defective block (one of the side "walls" of the slot is not perpendicular to the "floor" (surface 3) of the slot). A feature-based CAD system would allow the explicit representation of the slot, usually as a single high-level entity that is a component of the block object (most of the CAD packages actually implement features as instances of primitives* with several size and location parameters that can be varied). Thus, slotting operations could be directly verified by an automated visual in-process inspection system.

3.1.2 Defect Modeling

CAD models are also useful because they offer the potential of modeling common defects. It is likely that some of the possible part defects are known *a priori* and that they can be modeled geometrically [73]. If a part is defective, an automated inspection scheme can attempt to determine the nature of the defect by comparing the observed defect to the models of common defects. The generality of the CAD representation also allows new defects to be quickly introduced into the existing structure. It should be noted, however, that there do

*Primitive instancing is a popular solid modeling representation that is discussed further in Section 3.2.4.

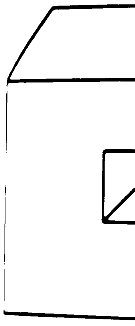


Figure 3.1. Defect-
faces labeled 1, 2, a
faces 1 and 2 are co

not appear to be
to be included in

3.1.3 Disadv

There are a few b
tion, however. O
usually not mode
model [132]. Fo
inspection and/or
are a tolerance of
degree polynomia
to compute exact
recognition, howe
original model [13
turing processes d
adjustments, but c
stage.

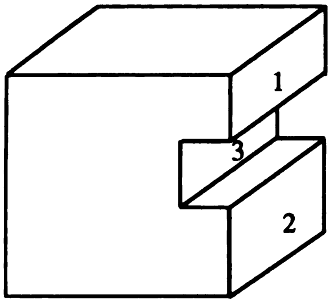


Figure 3.1. Defect-free block with slot. Surfaces labeled 1, 2, and 3 are parallel and surfaces 1 and 2 are coplanar.

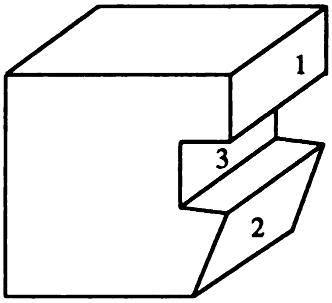


Figure 3.2. Block with defects. The slot is defective and the surfaces labeled 1 and 3 are parallel but 2 is not parallel to 1 or 3. Surfaces 1 and 2 are not coplanar.

not appear to be any current CAD packages which allow descriptions of potential defects to be included in the model of a part; defects must be modeled separately from the part.

3.1.3 Disadvantages of CAD Models

There are a few barriers to the widespread use of CAD models for automated visual inspection, however. One problem is that the process of image formation is very complex and is usually not modeled completely by the CAD model or by a supplemental image formation model [132]. For example, images often contain specular reflections, complicating the inspection and/or recognition tasks. Another problem is that computing offset surfaces that are a tolerance offset from curved surfaces is very difficult, requiring the solution of high degree polynomial equations [132]. For some surfaces, there are not any known methods to compute exact offsets. The greatest barrier to using CAD models for inspection or recognition, however, is that many times the actual manufactured part is different from the original model [132]. This difference arises due to adjustments made in the actual manufacturing processes during development. The CAD model should be updated to reflect these adjustments, but often there is no feedback loop from manufacturing back to the design stage.

3.2 Repr

Many solid repr
advantages and

Solid model
and representati
and volumetric
not use either of
representations.
extensions of d
contain no infor
restricted to the
are that interfere
and 3D objects c

The six mos
stancing, constr
representations [
The CSG and B-
B-rep and sweep
primitive instanc
techniques are hy

3.2.1 B-reps

The boundary re
using boundaries
organized into "s
are closed and co

3.2 Representation of Solids

Many solid representation methods have been proposed. Each of the methods has certain advantages and disadvantages that will be discussed in this section.

Solid models can generally be classified into two categories, representation by boundary and representation by interior [151]. Some researchers prefer the equivalent terms surface and volumetric representations, respectively. Many of the early geometric models did not use either of these two general solid modeling techniques, but instead used wireframe representations. Wireframes use edges to represent projections of solids and are simple extensions of drafting systems [151]. Wireframe models are ambiguous because they contain no information about the shape of the faces of the object, unless the class of objects is restricted to the convex polyhedra [151]. Other advantages of solid models over wireframes are that interference checking and mass property calculations can be accomplished easier and 3D objects can be described quicker and easier using solid models [161].

The six most common solid modeling techniques are the *B-rep, sweep, primitive instancing, constructive solid geometry (CSG), spatial enumeration*, and *cell decomposition* representations [151]. All of these techniques are unambiguous object representations [161]. The CSG and B-rep methods are the most common and best understood of these six. The B-rep and sweep methods are true boundary techniques while the spatial enumeration and primitive instances techniques are based solely on region interiors [151]. The other two techniques are hybrid methods.

3.2.1 B-reps

The boundary representation, or B-rep, method explicitly represents the object model using boundaries that are decomposed into sets of faces, edges, and vertices. Objects are organized into "shells" (closed connected sets of faces) and "loops" (face boundaries that are closed and connected edge sets) along with one or two vertices (endpoints) that bound each edge [116,

topological infor

that form the sol

and the equation

cannot be repres

than two [116, 1

A line drawin

is displayed in F

shown.

B-reps are id

because it is eas

corners, and othe

ciated with each

and inspection.

quire traversal o

milling [116]. Fi

accomplished mo

Many 3D obj

on the B-reps. Po

*For example, the
one (arbitrary) verte

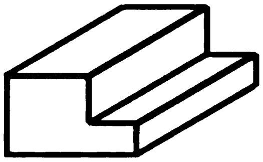


Figure 3.3. A line drawing of a block.

each edge [116, 151].[†] The B-rep contains topological and geometric information. The topological information describes the connections between the vertices, edges, and faces that form the solid object [161, 116]. The geometric information encodes vertex locations and the equations describing the edges and faces [161]. Often, surface intersection curves cannot be represented exactly, particularly when the face equations have degree greater than two [116, 151].

A line drawing of a block is shown in Figure 3.3. A portion of the B-rep for the block is displayed in Figure 3.4. The face entities and some of the topological information are shown.

B-reps are ideal for computer graphics and are also attractive for vision applications because it is easy to extract object surface information and object features such as holes, corners, and other salient features [111, 161]. The list of vertices and holes or openings associated with each face provide local features that may be useful for pose determination [161] and inspection. B-reps are also appropriate for numerically-controlled operations that require traversal of the outer boundary of the object, such as generation of instructions for milling [116]. Finally, local changes to the model, such as edge chamfering or filleting are accomplished more easily using the B-rep than other representations, such as CSG [116].

Many 3D object recognition systems in computer vision have used object models based on the B-reps. Polyhedral and piecewise quadric representations have been the most popular

[†] For example, the circular boundary of the planar "caps" of a cylinder is often represented as an edge with one (arbitrary) vertex. Most edges have two vertices, however.

B-reps i

3.2.2

The oth

The CSC

it is poss

implicitl

tive soli

each non

tion trans

analogs.

operation

The C

representa

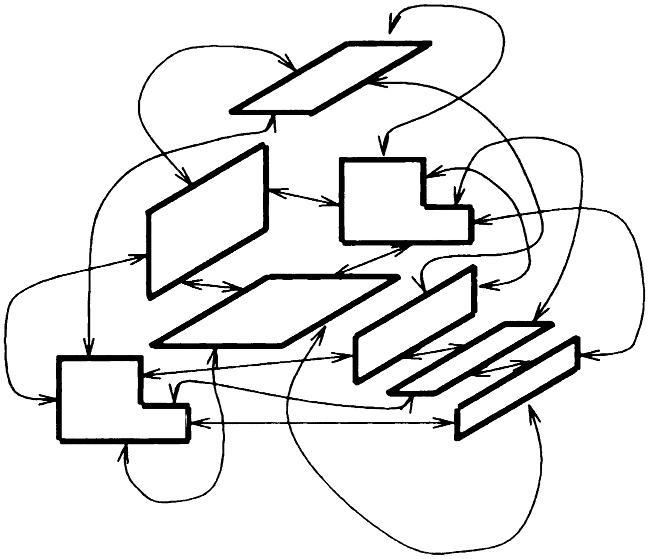


Figure 3.4. Portion of B-rep for a block.

B-reps in the computer vision literature [127].

3.2.2 CSG

The other most common representation is the constructive solid geometry (CSG) model. The CSG representation is usually a strictly volumetric representation of the part, although it is possible to include boundary information in the CSG model. A CSG representation implicitly represents an object using Boolean operations and transformations on primitive solid volumes. Usually a binary tree is used to describe the representation where each nonterminal node contains a Boolean set operation or a scaling, translation, or rotation transformation. Some of the construction operations in the tree have direct physical analogs. For example, one Boolean operation might represent a hole created by a drilling operation [151].

The CSG representation is non-unique, however (that is, there are many different CSG representations for the same object), and the CSG representation of objects cannot be

Figure 3.5. A C

directly compar

to create objects

non-trivial [111]

difficult [112].

that are difficult

discussed in mor

CSG represe

the fast, robust

synthesis of shad

to modify or dele

as straightforwar

One CSG rep

3.2.3 CSG vs

There are certain

been presented i

this discussion.

The current B-re

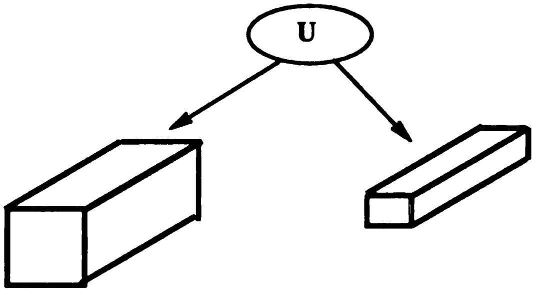


Figure 3.5. A CSG representation of the block of Figure 2.3. Union operation is shown.

directly compared. Instead, it is necessary to apply the operations in the representation to create objects that are compared [151]. The extraction of surface information is also non-trivial [111]. Finally, representing tolerance information in the CSG model is very difficult [112]. The principal difficulty with tolerances is caused by non-explicit datums that are difficult to determine and which result in tolerance stack-up. Tolerance stack-up is discussed in more depth in Chapter 9.

CSG representations offer several advantages. Many of the advantages arise from the fast, robust divide-and-conquer algorithms that operate on the CSG. These include synthesis of shaded images and computation of volumetric quantities [116]. It is also easy to modify or delete primitives or subtrees in the CSG. Deletion of operations is usually not as straightforward in the B-rep.

One CSG representation of the block of Figure 3.3 is shown in Figure 3.5.

3.2.3 CSG vs. B-rep

There are certain advantages to both the CSG and B-rep methods, many of which have been presented in the previous sections. Several additional issues are also germane to this discussion. One issue is the representation of manifold versus non-manifold objects. The current B-rep systems can only represent manifold objects [116]. The definition of

manifold o
can represe
regular set
whose edg
this is not a
during mod

Feature-
strictly bou
are usually
often involv
to encode i
design is m
usually do n
modeling se

Represent

There are se
algorithms t
there is no k
lutions to th
straightforw
fied [116]. A
an evaluated
use both the
which may "i

manifold objects requires that each edge belongs to only two faces. CSG representations can represent manifolds as well as any non-manifold that is also a regular set [116]. A regular set is an object whose faces separate solid material from background and all of whose edges form part of the face boundary. Since non-manifolds are not manufacturable, this is not a serious fault with the B-rep. However, non-manifolds can be easily created during modeling operations.

Feature-based modeling is another factor that complicates the decision to use either a strictly boundary-based representation scheme or to use a CSG representation. Features are usually created by performing some operation on existing portions of the boundary and often involve the creation of datum planes and reference lines [116], items that are difficult to encode in the CSG tree. But features also may have global effects, particularly if the design is modified in the future by modification of feature parameters [116]. B-rep models usually do not contain the information necessary to make such changes. Thus, feature-based modeling seems to require both a CSG tree and a boundary representation.

Representation Conversion

There are several known algorithms for converting a CSG representation to a B-rep. These algorithms tend to be very involved and numerically delicate, however [116]. Furthermore, there is no known algorithm to derive CSG from a B-rep. Thus, one of the apparent solutions to the feature-based modeling problem—using a dual representation—is not very straightforward. Thus, most dual-representation models allow only the CSG to be modified [116]. Another possible solution is to use a hybrid model where a CSG tree can contain an evaluated B-rep as a leaf node. This solution requires the design software to be able to use both the CSG divide-and-conquer approach and the B-rep trimmed surface approach, which may "introduce more conceptual and architectural problems than it solves" [116].

Free-form Su

The demands o
tions, resulting
representations.
ing the ability
of free-form su
recognition and

The class of
Coon's Patches
Uniform Ratio
quadric surface
are also invaria
projection [14(
ferent shapes c
NURBs are su
Standard (IGE
between most (

A NURB s

where w_{ij} are
direction and n
degree p and q
of $(n+p)$ non-

A key diffic

Free-form Surfaces

The demands of current manufacturing almost necessitate the use of boundary representations, resulting in most CAD packages using either B-reps or some form of hybrid or dual representations. The increasing use of plastics in industry is one of the key factors requiring the ability to represent complex surface geometries [116]. However, representation of free-form surfaces introduces additional complications for modeling and also for object recognition and inspection.

The class of free-form surfaces includes representations such as rational Bezier patches, Coon's Patches, Rational B-Splines, Gordon Surfaces, Algebraic Surfaces, and the Non-Uniform Rational B-Splines (NURBs). NURBs can exactly represent conic sections and quadric surfaces and contain the class of Bezier curves and surfaces [140, 171]. NURBs are also invariant under scaling, rotation, and translation and under parallel and perspective projection [140]. As is the case for other splines, NURBs offer design flexibility since different shapes can be designed by manipulating the NURB control points and weights [140]. NURBs are supported as an allowable representation within the Initial Graphics Exchange Standard (IGES), an ANSI standard data exchange format supported for data exchange between most CAD systems.

A NURB surface S with parameters u and v is defined as:

$$S(u,v) = \frac{\sum_{i=1}^{n}\sum_{i=1}^{m} w_{ij} B_{ip}(u) B_{jq}(v) \mathbf{P_{ij}}}{\sum_{i=1}^{n}\sum_{i=1}^{m} w_{ij} B_{ip}(u) B_{jq}(v)} \;,$$

where w_{ij} are the weights, $\mathbf{P_{ij}}$ are the 3-D control points with m control points in the u direction and n control points in the v direction, and $B_{ip}(u)$ and $B_{jq}(v)$ are the B-splines of degree p and q, respectively, defined on a knot vector. The knot vector for $B_{ip}(u)$ is a set of $(n+p)$ non-decreasing constants $\{u_0, u_1, ..., u_{n+p-1}\}$ that divide the evaluation domain.

A key difficulty with NURBs and other free-form surface representations is the computation of surf

quadrics, fre

such as the i

exactly [116]

intersections.

or piecewise

dering, but i

completely o

Blending

for free-form

blendings an

plexity of the

however, usi

and the offse

There are

representatio

Incorrect app

patch [140].

NURB patch

parameters o

stated in the

from the repr

parameters s

there can still

One obsta

[2] The *offset su*

from surface S.

$z = 0.1''$.

tation of surface intersections. While surface intersections can be computed exactly for the quadrics, free-form surface intersections require higher-order curves. Some intersections, such as the intersection between two bicubic parametric surfaces, cannot be represented exactly [116]. As a result, it is usually impractical, if not impossible, to represent surface intersections. The intersections are commonly approximated using meshes of planar facets or piecewise quadrics [116]. Approximating the intersections saves some time during rendering, but introduces additional complications since the intersection curve does not lie completely on both surfaces.

Blending, trimming, transformations, and surface offsets[‡] are also difficult to compute for free-form surfaces. If NURBs are adopted as the free-form surface representation, blendings and intersections can be implemented in a single algorithm, although the complexity of the resultant surface can be quite high. In general, surface offsets are not closed, however, using any representation; for example, the offset of a parabola is not a parabola and the offset of a NURB curve or surface is not even a NURB [116].

There are several other barriers to more widespread use of NURBs (and most other spline representations). For one, they require more storage than analytic curves and surfaces [140]. Incorrect application of the weights can also result in a poor parametrization of a surface patch [140]. To avoid degeneracies on some simple surfaces, such as a sphere, several NURB patches may be required to represent the surface [115]. Finally, the "natural" parameters of common surfaces, such as the radius of a sphere or cylinder, are not explicitly stated in the representation. Although it is technically possible to recover these parameters from the representation (though this may not always be feasible since recovery of "simple" parameters such as sphere radii requires the ensemble examination of several NURBs), there can still be round-off errors in retrieval [115].

One obstacle to the use of NURBs for computer vision purposes is that NURBs are

[‡] The *offset surface* S_o of a surface S is defined to be that surface which is a uniform distance (i.e., offset) from surface S. An example of a surface offset 0.1" from the planar surface $z = 0''$ is the planar surface $z = 0.1''$.

not tensor prod
data to the mo
in image data
quite difficult t
challenging to
patches are sm
patch begins a
surfaces. The
surface(s), no
make feature
geometric fea
of discontinui
recognition, a

In genera
surface matc
error toleranc
extract featu
is difficult. S
be overcom
not possible
Taubin [167
this distanc
approximat
parametric

not tensor product surfaces in three-space. This makes it very difficult to compare image data to the model's NURB representation. Determining where the surface knot points are in image data is very difficult and is an open research problem. It is also, in general, quite difficult to determine visually salient features on spline surfaces [110, 112]. It is also challenging to segment surfaces in image data of objects modeled by splines; since all the patches are smoothly joined (i.e., blended), it is not usually apparent where one surface patch begins and another ends [112]. In addition, there are no constraints on free-form surfaces. There need not be any piecewise-planar or piecewise-quadric regions on the surface(s), nor do they generally contain axes of symmetry [10]. These characteristics make feature extraction difficult and prevent the use of well-understood and often-used geometric features. Furthermore, free-form surfaces can contain cusps or isolated points of discontinuity in surface normals [10]. These issues make segmentation, classification, recognition, and verification difficult.

In general, 3D range sensors also do not deliver data accurate enough for free-form surface matching, particularly when the matching must be accurate within a very small error tolerance [10]. Inaccurate range data makes it more difficult to segment images and extract features for matching from the images; robust feature extraction and segmentation is difficult. Sensor technology is improving rapidly, however, so this objection may soon be overcome [10]. A more serious objection to the use of free-form surfaces is that it is not possible, in general to determine the distance of a point from a free-form surface [167]. Taubin [167] has presented a numerical method that finds a first-order approximation of this distance, although his method sometimes finds local minima that are not reasonable approximations of the global minimum. Matching and inspection of NURB and other parametric spline surfaces is thus a difficult task.

A Possible Matchi

Despite the diffi
surfaces in the
potential surface
been successfully
to be applied to
technique would
the model. If th
generate an acce
be attempted. F
matching could
includes a globa
and model surf

Besl has su
etc., could be
extracted inclu
and surface n
Gaussian cur
surface patch

There are
to Besl [10],
Also, spurio
extracted fro
are difficult
hierarchy se
the general

A Possible Matching Technique

Despite the difficulties with free-from surfaces, it may be possible to match free-form surfaces in the object to those in the model. Besl [10] has presented an outline of a potential surface matching algorithm that is theoretically possible but which has not yet been successfully implemented. He has suggested a hierarchical collection of techniques to be applied to the surface representation of the model and to the image data. The first technique would consist of trying to match point features extracted from the data and from the model. If that is not possible, or if not enough point features could be matched to generate an acceptable or complete correspondence, then matching of curve features could be attempted. Finally, if that also did not produce an acceptable matching, then surface matching could be attempted. After matches have been hypothesized, Besl's proposal includes a global verification step to compute the pointwise difference between data points and model surfaces.

Besl has suggested that cusps, vertices, isolated umbilic points, curvature extrema, etc., could be used as invariant point features for matching. Curve features that could be extracted include zero crossings of curvature, surface boundaries, curvature discontinuity, and surface normal discontinuity. For surface matching, Besl proposed the extraction of Gaussian curvature to partition smooth surfaces into elliptic, parabolic, and hyperbolic surface patches.

There are a few problems with Besl's proposal. One of the major problems, according to Besl [10], is that current sensor limitations prevent accurate estimation of curvature. Also, spurious point features are invariably extracted by feature extractors and curves extracted from image data are usually broken curves. Segmentation and feature extraction are difficult problems and are areas of ongoing research in computer vision. Finally, Besl's hierarchy seems to emphasize point features more than curve and surface features, despite the general consensus that surface parameters are more easily and robustly extracted from range data [58]

3.2.4 Othe

Samet [151] b
sweep represe
dimensional) s
components. A
circle swept al
vision, primari
adopted it for t
solid modeler t
nodes of the C

In primitiv
are created by
created by "att
viewed as an o
parameters are

Spatial enu
elements, also
would appear s

Cell decom

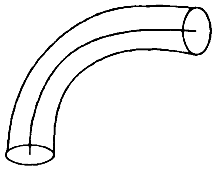


Figure 3.6. Example of a swept solid.

range data [58].

3.2.4 Other representations

Samet [151] briefly discusses the other four common solid model representations. The sweep representation is a solid that is defined by the volume swept by a planar (i.e., two-dimensional) shape along a curve. The sweep can have both rotational and translational components. An example of a swept solid is provided in Figure 3.6. The solid consists of a circle swept along an arc. The sweep representation has been commonly used in computer vision, primarily in the form of generalized cylinders, although few CAD systems have adopted it for their internal representation [55]. Mirolo and Pagello [119] have designed a solid modeler that combines CSG and generalized cylinders, however, by allowing the leaf nodes of the CSG tree to contain generalized cylinders.

In primitive instancing, all possible object shapes are defined *a priori*. Specific instances are created by changing scale and dimension parameters of the primitives. Objects can be created by "attaching" several of the primitives together. The block in Figure 3.3 could be viewed as an object consisting of two of a "rectangular box" primitive whose dimensional parameters are different in the two joined instances.

Spatial enumeration approximates the object using a collection of equal-sized volume elements, also called voxels. The spatial enumeration model of the block in Figure 3.3 would appear similar to the representation in Figure 3.7.

Cell decomposition is a hierarchical adaptation of spatial enumeration. In cell decompositi

are ty

repres

marke

space

the hal

of the

of the

Octre

Octrees

octree's

occupie

meaning

represen

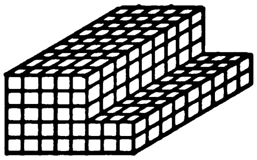


Figure 3.7. Example of spatial enumeration representation of block.

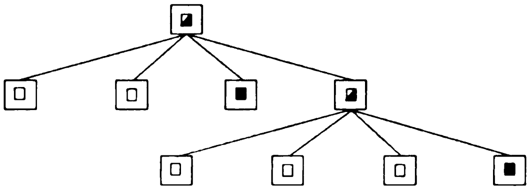


Figure 3.8. Example of cell decomposition representation of block.

position, the volume element sizes usually vary. Cells that are "higher" in the hierarchy are typically larger than the cells "lower" in the hierarchy. One cell decomposition tree representation for the block in Figure 3.3 is shown in Figure 3.8. In the tree, each node is marked with a solid, hollow, or half-solid square indicating if the node represents a region of space that is interior, exterior, or both interior and exterior to the object, respectively. (All of the half-solid cells are decomposed into smaller-sized volumetric cells in descendant nodes of the half-solid parent node.) The octree and its variants are probably the most common of the cell decomposition techniques.

Octrees

Octrees [77, 151] are a recursive hierarchical division of cubic space into subcubes. An octree's terminal nodes represent volumes of space that are either "full," (i.e., they are occupied by the interior of the object), or "empty," (i.e., they are occupied by "background" meaning that the node represents space exterior to the object). The root node of the tree represents the entirety of 3-space. Any node within the tree that is not homogeneous—that

is, any node th
which become
or a resolution
nodes, and no

Octrees ar
They are son
with which v
are useful fo
surface remo
avoidance an

A disadva
is that they d
tation is hig
Since octree
preserve the
the shape dis
limited class
resentations
fenders, vol
and time-co
Octrees do
features and
octree repre

is, any node that is not completely full or empty—is split into eight equal subcubes, each of which becomes a child of the node. This process continues until all nodes are homogeneous or a resolution limit is reached. Thus, an octree contains object leaf nodes, background leaf nodes, and non-leaf nodes.

Octrees are a pure volumetric description of the object that can be stored compactly. They are somewhat popular due to several additional advantages, including the speed with which volume elements can be located and their position determined [77]. Octrees are useful for graphics applications because they allow for easy rendering and hidden surface removal [77]. Octrees are also well-suited to robotics applications such as collision avoidance and path planning [77].

A disadvantage of the octree and, to a lesser extent, the spatial enumeration techniques, is that they do not provide an exact representation of the object. Furthermore, the representation is highly dependent on the position and orientation of the object in space [77, 151]. Since octrees have finite resolution, any rotational or translational transformation does not preserve the exact shape of the object. Several techniques have been proposed to minimize the shape distortion, however [77]. Octrees can also represent the exact geometry of only a limited class of objects [116]. In fact, octrees and other volumetric models are not ideal representations for many applications. For example, in the design of airplane fuselages and car fenders, volumetric characteristics are irrelevant and make modeling unnecessarily tedious and time-consuming [116]. The time to construct an octree can also be quite lengthy [151]. Octrees do not appear to be ideal for inspection purposes due to the difficulty of extracting features and surface information, although Tarbox and Gerhardt [166] are attempting to use octree representations for casting inspection purposes.

3.3 IDEAS

The IDEAS CAD
it is a popular co
projects in CAD-
experimental CAI
developed in this
applications. An
it allows construc
tasks. IDEAS' fe
One limitation is t
features can be c
only implied by t
inspection task.

IDEAS uses a
and a CSG-style h
One is a faceted a
to perform fast ca
CSG tree for an o
This tree can be
design intent. Fo
along some edge
object be changed
IDEAS to reflect t
some operation, th
percentage distanc

Extracting feat

3.3 IDEAS and IGES

The IDEAS CAD package is an attractive platform for building object models because it is a popular commercial package in industry. One drawback of many of the ongoing projects in CAD-based object recognition in the computer vision community is their use of experimental CAD systems. By using a commercial package such as IDEAS, the algorithms developed in this dissertation have more practical value for manufacturing and assembly applications. Another advantage of the current version of IDEAS (IDEAS VI) is that it allows construction of feature-based models which have some potential for inspection tasks. IDEAS' feature-based construction capabilities are somewhat primitive, however. One limitation is that tolerance information cannot be stored with the feature. Furthermore, features can be constructed such that their location with respect to a desired datum is only implied by the design rather than made explicit. This introduces ambiguity into any inspection task.

IDEAS uses a hybrid technique for solid modeling [102]. It stores both a precise B-rep and a CSG-style history tree for each object. Two B-reps are actually stored for an object. One is a faceted approximate representation that is used to generate displays quickly and to perform fast calculations. The other B-rep is a precise mathematical representation. The CSG tree for an object contains the history of the construction steps that created the object. This tree can be modified. IDEAS' CSG tree also supports the use of rules that define design intent. For instance, a through-hole can be defined to be a percentage distance *d* along some edge and at some fixed distance from the edge. Should the dimensions of the object be changed in the future, the position of the hole would be adjusted automatically by IDEAS to reflect the global change in the object. (For example, if the edge was scaled by some operation, the position of the hole would be adjusted such that it was still located at percentage distance *d* along the edge.)

Extracting features from the CAD model is not a simple and direct process. This is

because the CAD
but rather to perm
allowing designer
most CAD repres
about surface rela
are popular featu
representations al
and visibility. F
parameters from a

Flynn [55] has
created by the II
3D object recogn
bounding box ar
between surfaces
directly extracted
their values in sy
with several add
estimation. (Th
potentially usefu
the identity of th
included in Chap
to synthesize ran
scheme proposed

because the CAD model files were not designed for object recognition and inspection, but rather to permit many different CAD systems to read and write the same format files, allowing designers to easily port their designs from one system to another. For example, most CAD representations, including the IGES format, do not explicitly store information about surface relationships [59]. However, surface adjacencies and relative orientations are popular features for object recognition (matching models to sensed objects). CAD representations also do not usually contain viewpoint-dependent data such as surface area and visibility. Flynn and Jain [59] have presented techniques for estimation of these parameters from analytic-form (but not free-form) surfaces in CAD data, however.

Flynn [55] has written a LISP program to extract several features from the CAD models created by the IDEAS CAD package. Flynn [55, 59] used the following features for 3D object recognition: surface type and parameters, surface areas, visibility attributes, bounding box area, rotational validity, inter-surface orientational validity, and distances between surfaces. Except for surface area, bounding box area, and visibility, these were directly extracted from the CAD model. Flynn estimated the other features by observing their values in synthetic images generated from many viewpoints. These features, along with several additional features, are used in this thesis for object recognition and pose estimation. (The reader should note that although we present techniques that may be potentially useful for object recognition, the focus of this dissertation is inspection where the identity of the object is known.) Details about feature extraction and pose estimation are included in Chapters 5 and 6, respectively, of this dissertation. We also use the CAD model to synthesize range images from which a template is drawn for the gross defect detection scheme proposed in Chapter 7.

3.4 Repres

Inspec

Many of the requ

have been discus

desirable propert

representation w

and inspection,

representation of

of objects. Surfa

better suited than

3.5 CAD

The castings tha

package. Model

provide some C

models, and the

format, most CA

into the package

we used, which

The GM CA

and because they

result, there were

models from the

process, chiefly b

of the castings pr

3.4 Representation Requirements for Automated Visual Inspection

Many of the requirements for a representation useful for object recognition and inspection have been discussed in this chapter. Marshall [110, 111, 112] has outlined several of the desirable properties that 3D object models should possess for vision applications. An ideal representation would allow direct pairing of model and data features for efficient matching and inspection, direct estimation of pose, prediction of appearance from any position, representation of tolerances, and creation of a unique representation for any of a wide range of objects. Surfaces are compactly and explicitly represented by B-reps, so B-reps seem better suited than other representations for vision applications.

3.5 CAD Models of Castings

The castings that were inspected in this project were modeled using the IDEAS VI CAD package. Models were built based upon blueprints supplied by General Motors. GM did provide some CAD models in electronic format, but those models were only wireframe models, and they were stored in IGES format. Although IGES is an accepted exchange format, most CAD packages appear unable to completely translate all aspects of a model into the package's internal representation. This was certainly the case for the CAD package we used, which was unable to translate some portions of the GM models.

The GM CAD models were not usable because they were wireframes, not solid models, and because they only contained information related to set-up for the casting process. As a result, there were no dimensions stored in the GM IGES model. Thus, we had to build CAD models from the part blueprints. Model-building was a very tedious and time-consuming process, chiefly because of IDEAS' difficulties in creating filleted and rounded surfaces. All of the castings provided by GM contained many filleted and rounded surfaces, making this

a severe problem. Ther
subroutines. These bu
Another problem in r
that was often quite
necessary to first rem
followed by the mar
grew more complex
terminate. Some o
step, which usually
re-executes prior c
example, IDEAS n
or other primitives

Because of the
constructed a com
will demonstrate t
assume that simil
to demonstrate th
for inspection rat
extraction should

A drawing of
is shown in Figur
outline of this ge
A range image o
defect-free image
shown in Figure 4

A drawing of a

[1]The arufacts look

a severe problem. There are, in fact, known "bugs" in IDEAS VI's fillet and round creation subroutines. These bugs cause small polyhedral artifacts to appear on the filleted surface(s)[§] Another problem in model creation involved revising or correcting the model, a process that was often quite difficult. Usually, if a construction step s_c had to be corrected, it was necessary to first remove all the construction steps $s_i, i > c$. Step s_c could then be corrected, followed by the manual re-entry of construction steps $s_i, i > c$. Furthermore, as the model grew more complex, IDEAS became very, very slow and occasionally would unexpectedly terminate. Some operations also required IDEAS to re-execute every prior construction step, which usually took about 30 minutes of clock time on our system. (IDEAS probably re-executes prior construction steps to make the global model reflect local changes. For example, IDEAS must ascertain if the current operation will intersect any existing surfaces or other primitives.)

Because of the time-consuming nature of constructing a filleted CAD model, we actually constructed a complete CAD model for only one object, the gear blank 356 casting. We will demonstrate that we can extract the necessary inspection features from this model and assume that similar features can be extracted from the other models. This thesis aimed to demonstrate the feasibility of inspection, so the focus was on developing techniques for inspection rather than feature extraction from models. For the objects studied, feature extraction should be simple since no spline curves or surfaces were present.

A drawing of the CAD model generated in this research for the gear blank 356 casting is shown in Figure 3.9. The gear blank is shown from four different views. A more general outline of this gear blank with many of the dimensions labeled is shown in Figure 3.10. A range image of a defective instance of this gear blank was shown in Figure 1.1. A defect-free image from the top view is shown in Figure 4.9. A side and a bottom view are shown in Figure 4.4.

A drawing of a cylindrical casting is shown annotated with dimensions in Figure 3.11.

[§] The artifacts look like small bumps on the otherwise smooth surfaces.

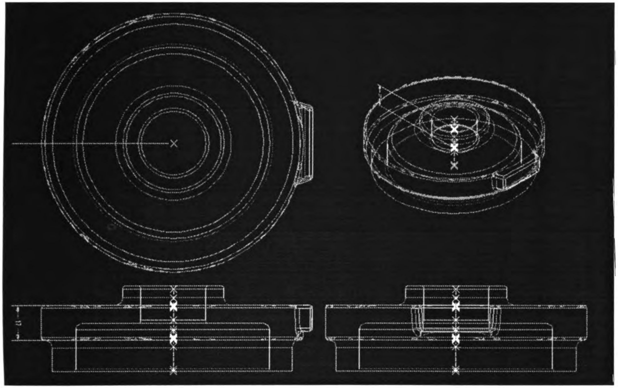


Figure 3.9. A wireframe rendering of the CAD model for gear blank 356.

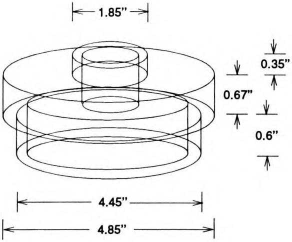


Figure 3.10. A simplified wireframe of gear blank 356 casting with dimensions.

Figure 3

When cast, this

the height dim

casting that G

of the gate was

region. This ca

casting is show

The drawin

casting will be

casting is show

shown in Figur

range image of

whose range im

The inspect

there is a CAD

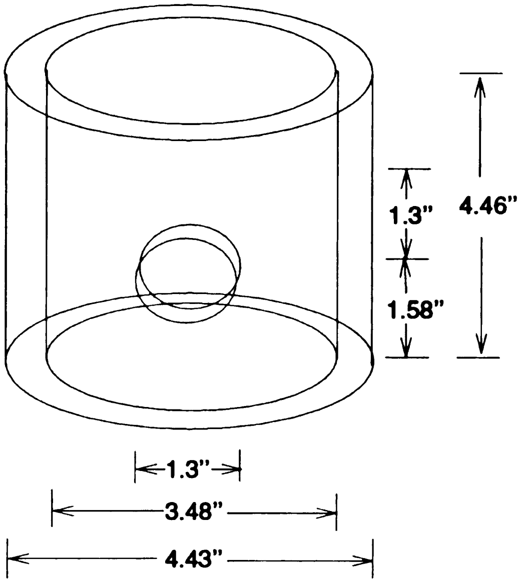


Figure 3.11. A simplified wireframe of a cylindrical casting with dimensions.

When cast, this casting also contains a gate that is about 1.5" wide and 2.2" in height, where the height dimension is in the direction of the cylinder's axis. All the instances of this casting that GM provided had nearly the entire gate removed, however. The only remnant of the gate was a region on the cylinder that was about 0.04" higher than the surrounding region. This casting will be referred to as "Cyl" in this dissertation. A range image of the casting is shown in Figure 4.8.

The drawing of a large cylindrical casting is shown in Figure 3.12. Instances of this casting will be referred to using the shorthand notation "Big Cyl." A range image of this casting is shown in Figure 4.7. A gear or pully blank casting, denoted number 588x, is shown in Figure 3.13. This casting will be referred to as "588x" in this dissertation. A range image of this casting is shown in Figure 4.6. Inspection models of two other castings, whose range images are shown in Figures 4.10 and 4.11, were also created.

The inspection techniques in this dissertation are applicable to any object for which there is a CAD model, although objects with non-quadric surfaces can only be inspected

Figure 3.12

Figure 3.1

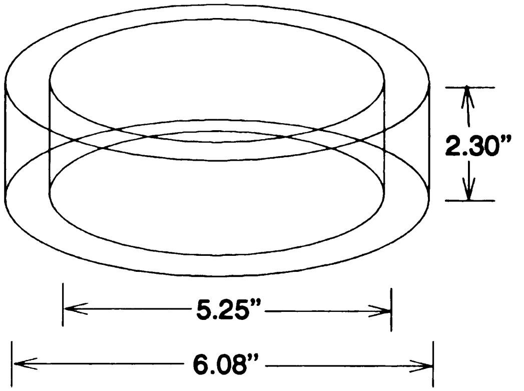


Figure 3.12. A simplified wireframe of a large cylindrical casting with dimensions.

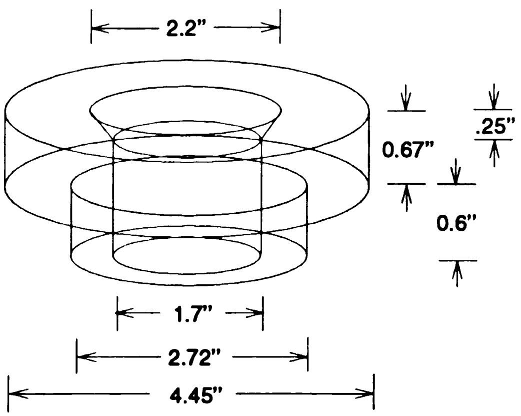


Figure 3.13. A wireframe of gear blank 588x casting annotated with dimensions.

if they are in
object pose is
spherical, and

3.6 Con

This chapter h
for 3D object
tions, CSG an
the common r
that allows for
and B-rep repr

This disser
only suitable C
both CSG and
applicable to c
however. It is
with filleted or
to inspection;
by IDEAS. Fu
inspection.

if they are in a known position (pose localization is discussed in detail in Chapter 6). If object pose is unknown, then our techniques allow the inspection of the planar, cylindrical, spherical, and some of the conical surfaces of any object.

3.6 Conclusions

This chapter has discussed the common representations used for computer-aided design and for 3D object recognition and inspection. In particular, the two most common representations, CSG and B-rep, were discussed in detail. The strengths and weaknesses of each of the common representation schemes were investigated. The best available representation that allows for both functions may be a hybrid or dual representation supporting both CSG and B-rep representations.

This dissertation uses CAD models built by the IDEAS commercial CAD modeler (the only suitable CAD package available to us) because it supports a hybrid representation using both CSG and B-reps. Because IDEAS is a popular commercial package, our work is more applicable to other inspection and recognition applications. IDEAS does have limitations, however. It is difficult to construct models of some objects in IDEAS, particularly objects with filleted or rounded surfaces. IDEAS also may not be mature enough for easy application to inspection; it may be difficult to generate an inspection plan for some objects modeled by IDEAS. Future versions of IDEAS or other CAD packages may be better suited for inspection.

CH

Ima

Video, or

source for

informatic

To circ

acquired di

most always

been decreas

Range data i

time-of-fligh

visible object

For compl

mation with ir

and similar str

is on geometric

modalities has r

potentially usefu

role of intensity

data source in thi

CHAPTER 4

Image Acquisition

Video, or intensity, images gathered by standard CCD cameras have been a popular data source for 3D object recognition and inspection. However, deriving 3D structures or depth information from a single 2D image continues to be a difficult and challenging problem.

To circumvent this problem, 3D data, called range or depth images, are now often acquired directly [92]. Unfortunately, the range sensing hardware is usually slower, and almost always more expensive than intensity sensing hardware. The cost of range sensors has been decreasing, however, while the quality of the acquired images has been improving [85]. Range data is now a preferred mode for 3D object recognition. Various triangulation and time-of-flight sensors have been built that collect three-dimensional (x, y, z) coordinates of visible object surface points.

For complete inspection of the object, it would be desirable to integrate range information with intensity and/or color (reflectance) data. This would allow labels, textures, and similar structures on object surfaces to be examined. In this dissertation, the focus is on geometric shape defects, and thus inspection based on color and intensity sensing modalities has not been investigated. As has been noted in Chapter 2, intensity data is also potentially useful for detection of cracks and pits, however. Chapter 9 briefly discusses the role of intensity data in detecting defects in castings. Range information is the dominant data source in this research, however.

obje

4.1

4.1.1

Range i

were pr

inspectic

requires t

that it is u

an object

inspection

by grease a

range data,

makes it eas

allows shape

In contrast, r

recognition sy

A range an

Figure 1.1. A s

In this figure, t

a defective inst

*A pseudo-inten

that are rendered in

the surface is oriente

a "perfect" photogra

imaged using an orth

This chapter presents some of the common techniques for range image acquisition for object recognition and inspection and discusses the sensing arrangement used in this thesis.

4.1 Range Sensing

4.1.1 Range vs. Intensity Imagery

Range images offer several advantages over intensity images. Many of these advantages were presented in Chapter 2. One example of the need for range images is dimensional inspection of parts that have etched height variances, warp, or reliefs. This type of inspection requires that depth be measured accurately [159]. Another advantage of range sensing is that it is usually not sensitive to ambient lighting. It is also usually much easier to extract an object from its background in range images than in intensity images. In the casting inspection problem, range images offer the additional advantage of being less affected by grease and surface scratches than intensity images. The greatest advantage in using range data, though, is that it explicitly represents surface information [92, 111]. This makes it easier to extract and fit mathematical surfaces. Explicit surface information also allows shape information to be used in developing efficient matching algorithms [111]. In contrast, matching based on edges, which is the practice in many inspection and/or recognition systems, discards much of the surface information [111].

A range and intensity image of one of the defective gear blank 356 castings was shown in Figure 1.1. A set of three additional images of the gear blank casting is shown in Figure 4.1. In this figure, the leftmost image (a) is a pseudo-intensity rendering of the depth map* of a defective instance of the casting. The defect is a small pit in the upper left portion of

*A pseudo-intensity rendering of the depth map displays the relative orientation of the surfaces. Points that are rendered in lighter shades are oriented nearly horizontally, while darker shaded points indicate that the surface is oriented almost vertically. The rendering is called pseudo-intensity because it looks similar to a "perfect" photograph of the scene in which the object is lit from above by a directional light source and imaged using an orthogonal projection onto the 2D image plane.

(a) Ps
dering

Figure 4
intensity

the castin

arrow has

and (c) are

sources. T

the defect i

region. Th

the importan

the lighting v

rough spots o

be difficult to

defects) from

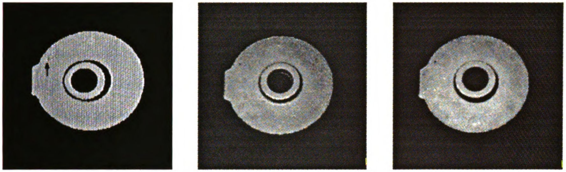
when histogram

could not be iso

casting's planar

defect in the rang

[1]Histogram-based



(a) Pseudo-intensity rendering of depth map.

(b) Intensity image with one light source.

(c) Intensity image using a different light source.

Figure 4.1. Three images of a defective gear blank 356 casting comparing range and intensity images.

the casting, at a position of approximately 10 o-clock near the outer circular boundary. An arrow has been overlaid on the image to highlight the defect. The images in Figures 4.1(b) and (c) are intensity images that are taken from the same viewpoint but using different light sources. The viewpoint is almost identical to that in the range image. In Figure 4.1(b), the defect is not very apparent, whereas in Figure 4.1(c), the defect is visible as a dark region. The two intensity images used light sources in different positions, illustrating the importance of lighting placement in intensity-based inspection. Furthermore, when the lighting was positioned such that the pit was more visible, some of the scratches and rough spots on the surface of the casting appeared as bright regions. In practice, it may be difficult to distinguish brightly colored regions that represent surface scratches (non-defects) from brightly colored regions that represent excess material (a defect). In fact, when histogram-based thresholding was applied to the image in Figure 4.1(b), the defect could not be isolated (its intensity is the same as many of the non-defective regions in the casting's planar surface) [†] The techniques presented in Chapters 8 and 9 can isolate the defect in the range image, however.

[†] Histogram-based thresholding did isolate the defect in the other intensity image, however.

d

or

Ar

(alt

rang

4.2

In this s

(i) trian

other ind

in [6, 55,

4.2.1 Tr

Triangulatio

literature an

(x, y, z) in th

procedure is

between the 3I

between the se

used to determi

In all triang

separation betwe

4.1.2 Depth Maps

A range image can be viewed as a depth map of data R_{ij}, where R is the range and (i, j) are the image coordinates in the x and y directions, respectively. Some sensing arrangements do not deliver this information directly and instead provide depth data in polar coordinates or simply provide a list of coordinates. The range sensor used in this research, the Technical Arts 100X White Scanner, produced a grid of depth values at regular intervals in x and y (although the grid was actually not completely regular in the y direction). Details about the range scanner are presented in Section 4.3.

4.2 A Taxonomy of Range Sensors

In this section, five of the common range sensing techniques are described. These include (i) triangulation methods, (ii) spot ranging, (iii) shape-from-X, (iv) ultrasound, and (v) other indirect methods. More detailed presentations of range sensing methods can be found in [6, 55, 130].

4.2.1 Triangulation

Triangulation techniques are quite popular for depth determination in both the research literature and the industrial vision applications [85]. In triangulation sensing, a point (x, y, z) in the 3D scene is projected to a point (u, v) in the 2D image plane. A calibration procedure is necessary in triangulation sensing to determine the geometric relationship between the 3D scene points and their 2D image plane projections. Usually, the relationship between the sensor and the light source is known *a priori*, allowing similar triangles to be used to determine the 2D-to-3D transformation for each image point.

In all triangulation systems, there is a tradeoff involved in the choice of the baseline separation between sensor(s) (and the illumination source if active illumination methods

Epp

are used

although

viewpoints

in fewer oc

multiple lig

and measure

is shown in F

Stereo

Stereo vision is

map for the poi

between two (or

be used to determ

The main prob

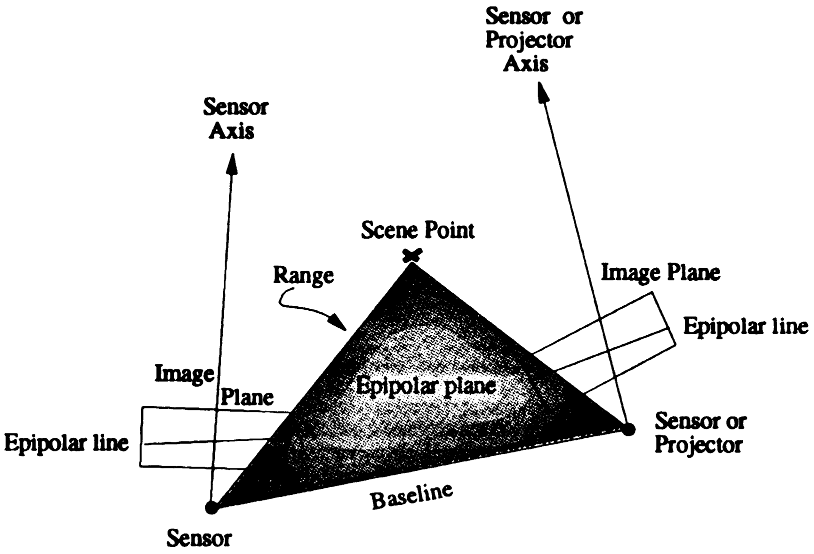


Figure 4.2. Diagram of triangulation sensing.

are used). If the baseline separation is large, depth can be estimated more accurately, although many surface points, especially in surface concavities, will not be visible from all viewpoints. Small baseline separations allow less accurate estimation of depth but result in fewer occluded surface points. This problem can be alleviated somewhat by the use of multiple light sources or multiple sensors [6, 130], although this also adds cost, complexity, and measurement time to the system [130]. A diagram of a typical triangulation arrangement is shown in Figure 4.2.

Stereo

Stereo vision is used by human and many other animate vision systems to find a disparity map for the points in an image. In stereo sensing, the known coordinate transformation between two (or sometimes more) sensors positioned at each of two triangle vertices can be used to determine the range values of the points in the image.

The main problem in stereo vision is the correspondence problem [130]. Matching, or *co*

Thi

of t

fairl

softw

slowl

maps i

De

Structu

Structur

light arra

is usually

project 2D

beam or lig

used before

by the light

Stockman [7

Structure

serially onto

patterns in pa

problem since

from multiple r

4.2.2 Spot R

Spot ranging met

years [6]. This cla

corresponding, the points in the two (or more) images needs to be done quickly and reliably. This process is complicated by missing data because some points are visible in only one of the images. Stereo is usually a passive process (i.e., it uses ambient light) that employs fairly inexpensive equipment, commonly consisting of only two intensity cameras. The software or hardware to compute point correspondences may be expensive and execute slowly, however [85]. Stereo methods also appear to be unable to generate dense depth maps in uniform (smooth) regions because of the difficulty in extracting features [85].

Determining range from motion stereo has also been investigated by Weng et al. [183].

Structured Light

Structured light techniques are low-cost active ranging methods. Most of the structured light arrangements can only produce sparse depth maps, however. A single plane of light is usually projected onto the scene, although techniques have also been developed that project 2D grids, coded stripes, or texture patterns onto the scene [85]. Usually, a laser beam or light is projected through a slit to illuminate the scene. A calibration procedure is used before image acquisition to establish the coordinates of the scene points illuminated by the light source. One example of a depth-from-structured light technique is by Hu and Stockman [79]. Their technique used one inexpensive light projector and a single camera.

Structured light methods tend to be slow, especially if the light stripes are projected serially onto the scene [130]. However, methods have been developd that project light patterns in parallel or by using time-sliced patterns. Specular reflections also present a problem since they can result in no depth measurement or in false depth measurements from multiple reflections.

4.2.2 Spot Ranging

Spot ranging methods for determining range have been used by civil engineers for many years [6]. This class of range sensors usually consists of a signal transmitter, receiver, and the

of th

orien

can d

quite

Th

uses a p

pulse. T

range (t

ambiguo

are emitt

Range is c

All of

several dif

no light bac

can make th

determined

slowly becau

4.2.3 Sha

The shape-from

intensity imag

Rather, they de

has been presen

orientation is obt

and the reflectan

photometric stere

the electronics for measuring round trip travel time (or its equivalent) and the intensity of the returned signal. This active technique usually utilizes a laser that is directed along orientations with equal angular increments using a scanning mirror. Spot ranging techniques can deliver a high density range map fairly quickly, although the cost of the sensor is usually quite high [85].

Three different techniques are used to measure the time of flight [130]. One method uses a pulsed signal and measures the actual pulse time delay for the roundtrip travel of the pulse. The phase shift of an amplitude modulated laser can also be measured to determine range (the phase shift is proportional to the time of flight). Phase shift determination is ambiguous, however, if the shift is more than 2π. The third method uses laser beams that are emitted as "chirps" that are frequency modulated as a linear function of time [130]. Range is determined by measuring the frequency-modulated signal.

All of these spot ranging techniques are accurate to within 1% of range [6]. There are several difficulties, however. One of the problems is specular surfaces that reflect little or no light back to the sensor. Alternately, multiple reflections from several specular surfaces can make the measured range appear much larger than its true value. Range cannot be determined reliably for such surfaces. The spot ranging techniques tend to measure range slowly because a long integration time is usually required to reduce photon noise [130].

4.2.3 Shape-from-X

The shape-from-X methods are a collection of techniques that determine depth using a single intensity image [85, 130]. Most of these techniques cannot determine depth explicitly. Rather, they determine surface orientation or relative depth. A wide body of literature has been presented on computing shape-from-shading. In shape-from-shading, surface orientation is obtained by examining image brightness, given the position of the light source and the reflectance properties of the object surface. A variation on this approach, called *photometric stereo* or *range from shadows*, uses several point light sources to illuminate thc

pos

and

som

sens

Spec

which

to illu

Sur

based c

based m

restricted

texture t

points.

4.2.4 U

Ultrasound

Ultrasound

reflected fro

Each surface

wave [6]. Du

tion about the

coordinates. N

only about a s

find internal pr

resolution than c

acoustic beam [1

the scene. The different light sources create sets of shadows that provide clues to the position of objects in the scene. Alternately, the object(s) can be rotated while the camera and light source remain stationary. Unfortunately, photometric stereo is not applicable in some scenes of solid 3D objects that are self-occluded. Self-occlusion prevents either the sensor or the light source from viewing or illuminating (respectively) many object points. Specular surface orientation has been determined using a variation on photometric stereo in which nonuniformly distributed light sources, such as circularly polarized light, were used to illuminate the object [130].

Surface orientation has also been determined using shape-from-perspective, a technique based on finding vanishing points. Orientation has also been determined using texture-based methods, although most of the shape-from-texture presentations in the literature are restricted to orientation determination for a single planar surface [14]. The shape-from-texture techniques usually determine orientation by fitting a line to a set of vanishing points.

4.2.4 Ultrasound

Ultrasound images are often used in medical imaging and in mobile robotics [39, 181]. Ultrasound sensors measure the time of flight of sound waves. The sound wave is often reflected from many surfaces, which results in a higher risk of incorrect range data [130]. Each surface encountered by the wavefront partially transmits and partially reflects the wave [6]. Due to the partial transmissions and reflections, ultrasound can provide information about the medium that makes up an entire column of data in a 3D matrix of 3-space coordinates. Most other methods, such as pulsed light beam ranging, provide information only about a single point in the scene [130]. This can be useful in medical imaging to find internal properties of a region of the body. Ultrasound images generally have a poorer resolution than other forms of range images because it is very difficult to generate a narrow acoustic beam [130].

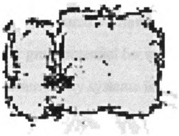


Figure 4.3. Ultrasound image of MSU PRIP Laboratory, from [39], courtesy of Jonathan Courtney.

An ultrasound image is shown in Figure 4.2.4, from [39]. This image is rendered as an occupancy grid map. Each cell in the map is of a fixed size and displays the measured intensity of ultrasound reflection in the MSU PRIP Laboratory, an indoor office-like environment. The darker cells have a higher probability of occupancy.

4.2.5 Other Indirect Methods

Several indirect passive methods for range sensing have also been used [130]. One of these methods is depth from focus. By focusing a lens and then finding the focal length and the distance of the image plane from the lens, the range can be determined. Another technique is range from known geometry which uses known camera parameters and known scene object identity and size to compute the range of other points in the image.

One other common technique for range sensing is called moiré interferometry, or depth from moiré fringes [130, 144]. The moiré fringe technique illuminates objects with light that has passed through a grating and views the scene through a camera that has been covered with an identical grating. The gratings produce a visible grid pattern, called moiré patterns or fringes, on the object surface. This pattern is a phase shift in the light that

r

to

ap

thr

15

alth

com

D

the d

4.3

In this

Image

range in

arrangen

that activ

The la

second of

triangular s

yields information about the change in object contours. Moiré fringe techniques have the advantage of allowing range to be measured on smooth surfaces. The techniques also tend to execute very quickly and offer great potential for rapid acquisition of depth values [46]. One disadvantage of moiré interferometry systems is that they are difficult to design and adjust [46].

Air Gage and the Industrial Technology Institute have jointly developed a range sensor based on moiré fringe techniques [165]. An experimental system they designed has a depth resolution of 1 micron and provides 14 bits of depth data through an image processing technique that is applied to the sensed data. A 512 × 484 depth map can be acquired in approximately 5 seconds. Air Gage has developed several commercial products based on this design, which offer a resolution of about 1.5 microns and can deliver a depth map every 15 seconds [21]. Their products have a field of view of approximately 3" by 3" by 1.5", although they have also developed custom designs for imaging automotive side panels. A complete systems retails for approximately $80,000.

Dubowsky et al. [46] have presented a moiré fringe technique that they used to inspect the dimensions of a diesel engine head gasket.

4.3 Range Image Acquisition

In this research, a Technical Arts 100X White Scanner in the Pattern Recognition and Image Processing (PRIP) Laboratory at Michigan State University was used to acquire range images. The presentation here follows that of Flynn [55]. A diagram of the sensing arrangement is shown in Figure 4.5. The White Scanner is a structured-light range sensor that actively illuminates a scene using a low-power 5 milliwatt helium-neon laser.

The laser beam is emitted from the laser tube and then reflected off two mirrors, the second of which is an optical spreader. It rapidly oscillates to create the appearance of a triangular sheet of light that illuminates objects within the work envelope of the scanner. The

sheet
envelo
10 to 1

The
directio
slightly
or smalle
Each tim
sheet of l
camera ar
the final ra
rotated abo
be collecte
available to
viewpoints

The sens
is oriented a
of each point
y and z coord
points. (A c
session to allo
row number in
CID image.) T
data to yield an
The range imag
stage movement
intervals in x (the

sheet of light is perpendicular to the x-y plane of the sensor coordinate system. The work envelope is a grid of maximum size 12" × 12" with a maximum height of approximately 10 to 12 inches.

The object(s) to be imaged are placed on a translation stage that is moved in the x direction using a stepper motor. The stage is usually moved 240 times over 6 inches (a slightly larger or smaller range of movement can be used when acquiring images of larger or smaller objects, respectively, allowing the capture of slightly higher resolution images). Each time the stage is moved, a 2D silhouette image is formed by the intersection of the sheet of laser light with the object(s) to be imaged. Each of the silhouettes is imaged by a camera and stored in the 100X computer. The collection of silhouettes is then merged into the final range image. The Laboratory also has a rotary stage that allows the object to be rotated about the z axis. If the object is imaged following rotation(s), multiple images can be collected, simulating a multiple viewpoint mode of image acquisition. No method was available to turn the object over, however. Thus, an object could only be scanned at known viewpoints in one viewing hemisphere.

The sensor is a CID camera that has a 248 × 388 grid of sensing elements. The camera is oriented at a fixed angle of 45° (about the y-axis) to the sheet of light. The x coordinate of each point in the scene is determined using the movement of the stepper motor while the y and z coordinates are determined using their position with respect to several calibration points. (A calibration procedure must be performed at the beginning of each scanning session to allow these points to be determined. The y coordinate is proportional to the row number in the CID image while the z coordinate is proportional to the column in the CID image.) The sensing arrangement delivers a resampling of the original 248 rows of data to yield an image that has 240 points of information at each of the imaging steps. The range image is formed by joining each of the rows of information associated with the stage movement. This results in a depth map $z = f(x, y)$ that has values at evenly spaced intervals in x (the direction of stage motion) and at nearly evenly-spaced values in y.

be v
in z
0.002
specif
the obs
about t
Scanner
is possib
a focusin
project.

Not al
self-occlus
shown in F
Black regio
represent sh
not contain
black and/or

The scann
have excellen
environments

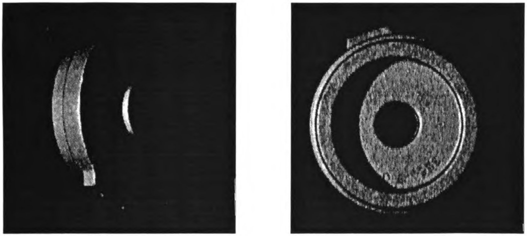
The sensin
the intensity in
intensity visible
similar in appear
the White Scann
and intensity ima

Sensor noise is present in both the y and z coordinates. The error in y appears to be very small but systematic, although its nature has not been characterized. The error in z appears to follow a Gaussian distribution with a standard deviation of approximately 0.002" to 0.005", both in tests performed by us and by those reported elsewhere. The vendor specification for the device claims that depth error is between 0.002" and 0.005" [170], so the observed behavior is similar to the expected behavior of the scanner. (More information about the sensor error is found in Chapter 8.) The depth values obtained by this White Scanner are estimated to 0.001". According to the White Scanner's reference manual, it is possible to achieve accuracies of "better than a few one ten-thousandths of an inch" if a focusing lens is fitted to the laser [170], although such a lens was not available for this project.

Not all pixels contain range information due to shadowing effects caused by concavities, self-occlusion, specularities, or noise. Two images of one of the gear blank 356 castings are shown in Figure 4.4 to illustrate typical problems with occlusion from the White Scanner. Black regions in the image represent points for which depth could not be determined. These represent shadows from occlusions or background pixels. Background pixels usually do not contain any range information because the stage and supporting devices are painted black and/or covered with a black cloth to minimize unwanted reflections.

The scanner is usable at temperatures between $-18°$ and $50°$ Celsius and is reported to have excellent immunity to vibration [170]. It is also usable in variable ambient lighting environments [170].

The sensing arrangement produces registered range and intensity information, although the intensity information is the value of the reflected laser beam rather than the usual intensity visible under regular indoor or natural lighting. The intensity data is somewhat similar in appearance to pseudo-intensity renderings of range images. Figure 4.4 displays the White Scanner intensity data for two images. It is also possible to collect fused range and intensity images from the White Scanner using Lee and Stockman's [103] imaging



(a) Side view of gear blank 356 casting.

(b) View from underneath gear blank 356 casting.

Figure 4.4. Two self-occluded images of gear blank 356 casting. The White Scanner's intensity map is displayed in both images.

procedure, although the procedure is unable to fuse any point that is not visible in either the range or the intensity image.[‡] This dissertation has focused primarily on range images, although the feasibility of using fused range and intensity images has also been investigated.

One limitation of the White Scanner is that it takes about five minutes to acquire the depth image of an object. Another problem in using the White Scanner is that while each point has a very accurate x, y, and z coordinate value, the sampled points in the y direction are separated by approximately 0.045" while the sampled points in the x direction are typically separated by about 0.025". Thus, the accuracy of the system is limited more by sample spacing than by measurement accuracy. Object surface points located near depth discontinuities are also often not sensed properly because the measured depth values could be the mean of the depth values of two points on either side of the edge [170].

Range images of some of the castings collected using the PRIP lab's 100X White Scanner are shown in Figures 4.6 through 4.11. These images are defect-free instances of

[‡] The intensity image in Lee and Stockman's procedure is not formed from an orthogonal projection, unlike the White Scanner's intensity image. The potential visible surface area is thus much larger in their procedure.

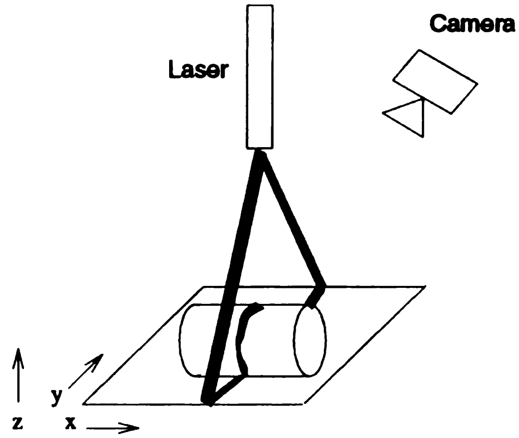


Figure 4.5. Diagram of Technical Arts Range Image Acquisition System.

the castings examined in this thesis. A depth profile plot of the gear blank 588x casting of Figure 4.6 is shown in Figure 4.13 and a depth profile plot of the spindle-like casting of Figure 4.11 is shown in Figure 4.12.

In addition to real range images, synthetically-generated range images have also been used in this project. Some of the synthetic range images were generated by sampling the z coordinates of quadric surfaces at regular intervals over a range of x and y coordinates. Other range images were synthesized from CAD models using a program that was originally based on Paul Besl's GEOMOD JUNIOR. Pat Flynn adapted GEOMOD JUNIOR to read IDEAS IV CAD models and we then extended it to read IDEAS VI CAD models. Synthetic range images were used to allow Monte Carlo testing of both defect-free and defective objects in many poses, allowing the capabilities of the algorithms to be explored in detail.

4.4 Summary

This section has presented some of the issues involved in data acquisition for 3D object inspection. The motivations for using range data rather than intensity data were discussed. The common techniques for acquisition of range data for 3D object recognition and inspec-

Fi
int
dep
588

Figure
intensi
depth n
gear 35

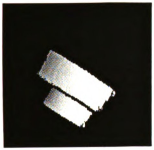


Figure 4.6. Pseudo-intensity rendering of depth map of defect-free 588x casting.

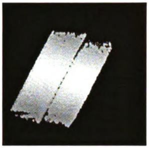


Figure 4.7. Pseudo-intensity rendering of depth map of defect-free large cylindrical casting.

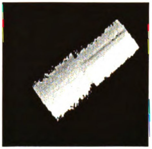


Figure 4.8. Pseudo-intensity rendering of depth map of defect-free small cylindrical casting.

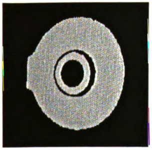


Figure 4.9. Pseudo-intensity rendering of depth map of defect-free gear 356 casting.

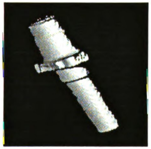


Figure 4.10. Pseudo-intensity rendering of depth map of defect-free housing casting.

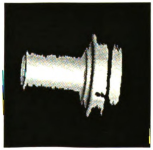


Figure 4.11. Pseudo-intensity rendering of depth map of defect-free spindle-like casting.

Figu

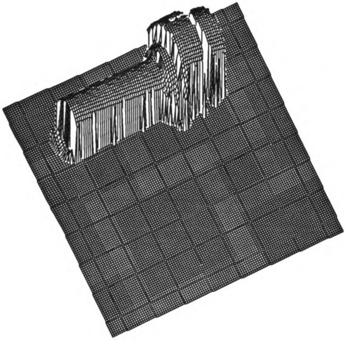


Figure 4.12. Depth profile plot of the spindle-like casting.

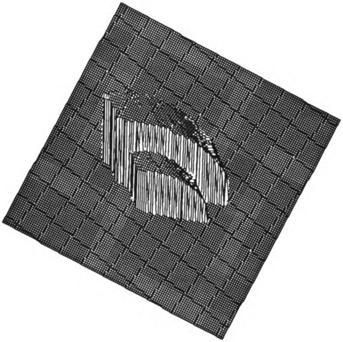


Figure 4.13. Depth profile plot of the gear blank 588x casting.

c

s

w

d

tion were also presented. Finally, the range image acquisition system used in this research was discussed.

The Technical Arts scanner available for this research can deliver accurate depth information, although its accuracy may not be acceptable for some high precision tolerance measurements. A more serious problem is that the scanner cannot inspect deep concavities in objects. Furthermore, self-occlusions become a difficult problem due to the large baseline separation between the sensor and the active light source. The available sensor is also too slow for real-time inspection applications, although other sensors have been reported that appear to meet (or nearly meet) this goal [165].

The problem of self-occlusions can be limited somewhat by collecting multiple images of the object from different viewpoints, however. This could be accomplished using rotating stages and fixtures to hold the objects. In this work, equipment was available to rotate lightweight objects about one axis, allowing collection of data from one "side" of the object. It is, in general, better to move the sensors than the object, however, since some objects are fragile. Ideally, the sensor would be mounted on a robot arm to collect multiple viewpoint data. Alternately, multiple sensors could be placed about the viewing area to collect nearly complete 3D data from the object to be inspected.

In the remaining chapters of this dissertation, the techniques for inspection are presented. Many of these techniques assume that the input data are range images collected with a triangulating sensor and thus include "corrective" action to overcome some of the deficiencies of the scanner.

C

S

Thi

ima

into

corr

the p

surfa

of th

in nea

purpos

realiza

compli

In th

of Flyn

Jain [75

and $(x, y$

merging.

also deli

Classi

CHAPTER 5

Surface Segmentation and Classification

This chapter discusses procedures for extracting and classifying surface regions from range images. The process of extracting surfaces, called segmentation, groups range pixels into homogeneous regions. The goal of the segmentation process is to group range pixels corresponding to different types of surfaces into different regions. The process of recovering the parameters of surfaces and subsequently labeling the region types in the image is called surface classification. Segmentation and classification are useful in determining which of the model surfaces are present in the image. They are important preliminary steps in nearly every object localization, recognition, and inspection paradigm. For inspection purposes, segmentation isolates surfaces and allows them to be compared against their ideal realization. Surface defects can sometimes confuse segmentation schemes, however, and complicate subsequent processing.

In this dissertation, segmentation was accomplished using an enhancement of the method of Flynn [55] which itself is an extension of the method first proposed by Hoffman and Jain [75]. The Flynn, Hoffman, and Jain algorithm performs a clustering of surface normals and (x, y, z) coordinates at every range pixel followed by a refining step of filtering and merging. This method is designed for images containing quadric surfaces but appears to also deliver adequate segmentations of some images containing other surface types.

Classification of surfaces into planar, spherical, cylindrical, or conical regions was

r

a

5

Se

reg

1.2

seg

A

can

Ther

to se

imag

surfac

homog

proces

produc

An

in Figu

algorithn

accomplished using model-driven techniques utilizing geometric characteristics of the surfaces and known model parameters, and based on linear fitting and Hough accumulation techniques. The planar, cylindrical, spherical, and conical surfaces are probably the four most important surface types for inspection because most manufactured parts are bounded solely by these surfaces [111].

In this chapter, we first discuss approaches to range image segmentation and our technique for segmentation of range images of castings. We then discuss surface classification and our model-driven technique for quadric surface classification.

5.1 Segmentation

Segmentation can be defined as a process of grouping pixels to form "homogeneous" regions or surfaces. Specifically, in an image $I(x, y)$ composed of surfaces $\{S_i, i = 1, 2, \ldots, n\}$, with $p_j \in I$, where $p_j, j = 1, 2, \ldots, m$ are pixels in the image, $m = \| I(x, y) \|$, segmentation is the process of assigning one of the labels S_i to each pixel p_j.

However, it is unlikely that any currently available image segmentation technique can provide a perfect segmentation for a wide variety of objects and imaging geometry. Therefore, it is necessary for matching, localization, and inspection schemes to be robust to segmentation errors. It is generally easier to recover from over-segmentation of an image (where the number of segments extracted is larger than the true number of visible surfaces on the object) than under-segmentation since over-segmented regions tend to be homogeneous whereas under-segmented regions exhibit heterogeneity, making subsequent processing much more difficult. Many of the segmentation schemes in the literature aim to produce over-segmentation in images.

An example of a range image with correct, over-, and under-segmentations is shown in Figure 5.1. The under-segmentation in Figure 5.1(d) is the result of Flynn's [55] algorithm following the iterative merging step applied to the over-segmented image of

(c)
fitt

Fig

con

with

in th

cause

comp

segme

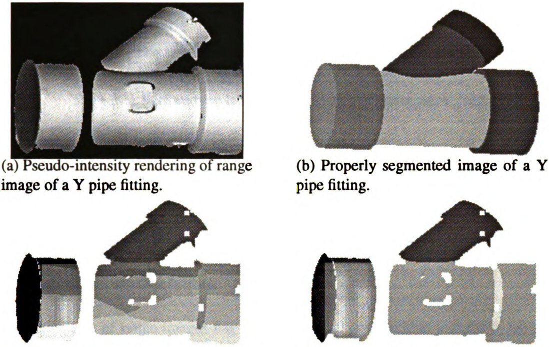
about

image

occur

algorith

*Chap



(a) Pseudo-intensity rendering of range image of a Y pipe fitting.

(b) Properly segmented image of a Y pipe fitting.

(c) Over-segmented image of a Y pipe fitting.

(d) Under-segmented image of a Y pipe fitting.

Figure 5.1. Range image with correct, over-, and under-segmentations.

Figure 5.1(c). One artifact to note is that the over- and under-segmented images both contain some missing data. This is because the range image in Figure 5.1(a) was captured with a triangulating range scanner and, as a result, self-occlusions caused shadows to appear in the image*. The depth data is indeterminate in these shadow regions. Sensor noise also caused unreliable range measurements at some surface points. Several one- or two-pixel components on the object's surface thus have no range value associated with them. Our segmentation scheme does not label those pixels or pixels within a small neighborhood about them. In contrast to the over- and under-segmented images, the correctly segmented image shown in Figure 5.1(b) was generated synthetically. Segmentation errors usually occur due to problems in the imaging process and due to problems in the segmentation algorithm (although most segmentation schemes do perform better on synthetic data than

*Chapter 4 discussed triangulating range sensors and some of the issues particular to the 100X scanner.

b

a

cl

th

ha

On

wh

to si

of "a

segm

et al.

S

quite

vision

	$K > 0$	$K = 0$	$K < 0$
$H > 0$	Pit	Valley	Saddle Valley
$H = 0$	Not possible	Plane	Minimal Surface
$H < 0$	Peak	Ridge	Saddle Ridge

Table 5.1. Surface type labels from surface curvature signs, from [12].

on noisy data).

5.1.1 Literature Review

Region-based segmentations

Segmentation of range images has been an area of considerable interest in the research community. In some 3D vision systems, features of a region are extracted and a ***region-based segmentation*** approach is followed. Region-based range image segmentation schemes have been developed by many researchers, including Besl and Jain [12], Faugeras et al. [50], and Hoffman and Jain [75]. Jain and Hoffman [89] grouped pixels using a squared error clustering on point coordinates and normal vectors to form surface patches and then merged the patches using knowledge of objects in a model database. The split-and-merge paradigm has been very popular in the literature for segmentation of both range and intensity images. One example of a split-and-merge intensity image segmentation scheme is that of Gay [61] who used both region and edge data to initially over-segment the intensity image and then to subsequently merge segments iteratively. Gay performed merging by computing forces of "attraction" and "repulsion" between segments. Marshall [110] adapted this technique to segment range images, although he apparently did not use any edge information. Faugeras et al. [50] segmented surfaces into planar or quadric patches using a region-growing method.

Segmentation methods based on the signs of Gaussian and mean curvatures have been quite popular. Since estimated image curvature values are very sensitive to noise, most vision researchers have instead used the sign of two curvature measures. Table 5.1 shows the

n
M
p
d
n
a
d
n
th

ea
al
cu

m
di
in
co
ar
th
su
tec
an
no

meanings of the combinations of curvature signs for mean (H) and Gaussian curvature (K). Mean curvature is the mean of the two principal curvatures and Gaussian curvature is the product of the two principal curvatures. The minimum and maximum principal curvature directions refer to the orthogonal directions in which curvature attains its minimum and maximum values, respectively [118]. The principal curvature values are these minimum and maximum values. Curvature is a measure of the bending of a surface. Curvature is defined as $\kappa(s) = \frac{\mathbf{T}'(s)}{\mathbf{N}(s)}$, where $\mathbf{T}(s)$ is the tangent vector to a curve $\alpha(s)$ and $\mathbf{N}(s)$ is the normal vector to the curve. Curvatures are attractive for use in computer vision because they are invariant to translations, rotations, and parametrization changes.

Besl and Jain [12] generated an initial coarse segmentation of range images by labeling each region with one of the eight possible combinations of H and K curvature signs. Ferrie al. [52] have presented a representation scheme for 3D objects based in part on principal curvatures.

In applying curvature-based approaches to computer vision tasks, several practical issues must be addressed. The major concern is the method of estimating the curvature from a discrete set of data (pixels). Since curvature describes the local behavior of a surface, it is important that pixels in only a small neighborhood around the pixel of interest be used in the computation of curvature. However, use of too small of a region in the image can result in an unreliable estimate of curvature. Curvatures are second-order differential properties and thus are inherently more sensitive to noise in the input image than first-derivative features such as surface normals. Flynn and Jain [56] reported difficulties in using curvature-based techniques to fit conic parameters to synthetic data even in the presence of only small amounts of Gaussian noise. The general consensus in the literature is that range sensors do not yet deliver data accurately enough for curvature-based approaches to be reliable [10].

c

u

fc

m.

fin

obj

Hyl

Hyb.

segm

edge

signs

tions v

discont

integrat

Seve

proaches

tion by co

Edge-based Segmentations

Another technique for range image segmentation is an *edge-based segmentation*. Bolles and Horaud [17] and Fan et al. [51] have developed object recognition systems that have utilized edge-based segmentation schemes. Fan et al. [51] presented a system for recognizing 3D objects that used curvature sign changes and extremal values as the basis for segmentation of range imagery. Bolles and Horaud [17] developed the 3DPO object recognition system and applied it to several images, including those of castings. 3DPO extracts circular and linear edge segments from range images and then recognizes objects by comparing these edges to lists of circular edges extracted from an extended CAD model. 3DPO's matching technique is similar to the local-feature-focus method of Bolles and Cain [16], because a few likely feature clusters are used to form an initial match hypothesis. Additional feature matchings that are consistent with the initial matching are then added to the match set and, finally, a least squares technique is used to compute the precise transformation between object and model.

Hybrid Segmentations

Hybrid segmentation schemes that combine region-based and edge-based approaches to segmentation have also been presented. Yokoya and Levine [192] used both surface and edge features for segmentation. They used the eight possible combinations of curvature signs to generate an initial region labeling. In addition, two initial edge-based segmentations were generated using depth discontinuities (i.e., jump edges) and surface orientation discontinuities (i.e., roof or crease edges). These three initial segmentations were then integrated into one final segmentation.

Several other segmentation methods that integrate region-based and edge-based approaches have been proposed. Jain and Nadabar [90], for instance, performed segmentation by combining clustering-based region segmentation with a Markov random field model e

s

n

O

wl

tru

hist

con

inter

objec

label

T

and D

vertic

of ran

the im

(i.e., po

techniq

non-ver

that rem

technique for edge extraction.

Segmentation of Fused Images

Several methods for segmentation of integrated range and intensity images have been presented in the literature. Sensor integration (fusion) is desirable for several reasons. Perhaps the most fundamental of these is that sensor fusion can allow more robust feature extraction than is possible when using a single sensor. Additionally, a sensor fusion model seems to be favored by natural vision systems. Mitiche and Aggarwal have observed that nearly all higher life forms use a variety of sensors to solve basic survival problems [120]. One example of a hybrid segmentation scheme was presented by Nadabar and Jain [122], who detected and labeled edges by fusing intensity and range images.

Jain et al. [91] have extracted regions containing vehicles (tanks, APCs, jeeps, and trucks) and calibration plates from registered range and intensity images by building a histogram of intensity values at regular range intervals. Each bin in the histogram is constructed by summing the intensity of all image points that are within that bin's range interval. The three highest peaks in the histogram are taken as the range values at which objects of interest may be present. Regions are then extracted by connected component labeling of pixels whose range fell within the interval of each of the peak bins.

The method of Jain et al. is related to the work of Nitzan, Brain, and Duda [131] and Duda, Nitzan, and Barrett [47]. Nitzan, Duda, et al., sequentially identified horizontal, vertical, and then arbitrarily-oriented planes in a range image. They constructed a histogram of range values whose peaks were used to extract all points lying on horizontal planes in the image. Vertical planes were found next by projecting all the remaining image points (i.e., points not lying on horizontal planes) onto a horizontal plane and then using a Hough technique to detect lines in the plane. In [47], arbitrarily-oriented (i.e., non-horizontal, non-vertical) planes were found by building intensity histograms of connected components that remained in the image after extraction of all detected vertical and horizontal planes. A

c

n

o

re

po

ite

tio

bou

0.4

of o

whi

plane was fit to each set of pixels whose intensities were near peaks of this histogram.

Chu et al. [37] have presented a segmentation scheme for extracting vehicles from registered range and intensity images. Their method segmented the range and intensity images separately and then integrated the segmentations. Planar surface fitting was used to segment the range image. Image statistics (mean and standard deviation) were used to segment the intensity image. Chen and Jain [29] have also extracted vehicles from registered range and intensity images. Their method was based on background removal. Verly et al. [179] have extracted and recognized military vehicles in identically registered range and intensity images produced from infrared-radar range imagery. They integrated contour primitives, such as corners and arcs from the object silhouette, and region primitives, such as region labels and inter-region relationships, in their scheme.

5.1.2 Range Image Segmentation Methodology

Flynn-Hoffman-Jain Segmentation

The segmentation scheme we used for extracting homogeneous regions from range images of castings is a refinement of the method of Flynn [55] and Hoffman and Jain [75]. Flynn's method used the surface labelings from Hoffman and Jain's algorithm as its initial estimate of segmentation. Flynn then filtered the segmentation and merged some of the adjoining regions. His filtering step consisted of first re-labeling the image using a connected component algorithm and then removing all small regions containing less than 20 pixels. An iterative scheme was used to merge adjacent regions provided that their surface orientation, as measured by the average change over all boundary pixels, varied little across the boundary. Merging was only allowed for adjacent regions whose difference in depth was 0.4" or less. (This tolerance delivered acceptable results for all of Flynn's model database of objects.) Next, regions with similar curvature features were merged. Finally, any region which contained less than 100 pixels after the merging process was removed.

(

r

v

0

o.

en

im

nev

New Segmentation Technique

The technique used for segmentation of casting images is a revision of Flynn, Hoffman, and Jain's method. It was necessary to revise their method because it did not deliver acceptable segmentation results for the casting images. An example of Flynn, Hoffman, and Jain's segmentation of gear blank image A-1 is shown in Figure 5.3(a). Clearly, the natural roughness of the castings created problematic segmentations. One problem was that too many small regions were removed in early stages of the segmentation, producing images with many unlabeled pixels. Their technique also assumed that small segments that deviated slightly from the surrounding data were composed of noisy measurements. Thus, many small clusters were not labeled by their scheme.

The Flynn, Hoffman, and Jain segmentation technique was actually not designed for inspection applications. Rather, it was designed to label regions for recognition purposes. An object can be recognized even if not every pixel can be labeled. But inspection requires more complete utilization of data. Small defects, for example, can only be found by examining all of the data.

We enhanced the Flynn-Hoffman-Jain algorithm in the following ways. First, we did not remove small regions until after the merging step based on surface orientation was carried out. After the merging, regions with less than 10 pixels are removed. Additionally, the merging tolerance value is much more conservative in the current segmentation. Merging was only conducted for adjacent regions with a difference in jump boundary of less than 0.1". These enhancements resolved many of the problems of unlabeled points on the interior of surface regions. A block diagram of the segmentation algorithm is shown in Figure 5.2.

A comparison of segmenting an image with the Flynn-Hoffman-Jain method versus our enhanced technique is shown in Figure 5.3. Our segmentation of gear blank 356 casting image A-1 is shown in Figure 5.3(c). Another comparison is shown in Figure 5.4. The new segmentation discards fewer points on the interior of the regions and near the region

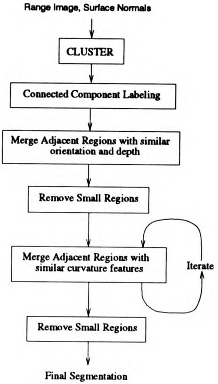
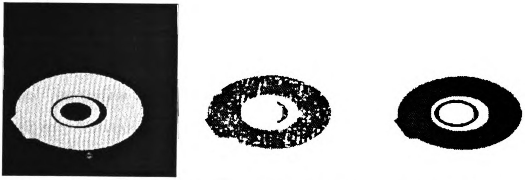

Range Image, Surface Normals
CLUSTER
Connected Component Labeling
Merge Adjacent Regions with similar orientation and depth
Remove Small Regions
Merge Adjacent Regions with similar curvature features
Iterate
Remove Small Regions
Final Segmentation

Figure 5.2. Block diagram of segmentation process



(a) Range Image of defect-free gear blank 356 casting.

(b) Flynn, Jain, and Hoffman Segmentation.

(c) New Segmentation.

Figure 5.3. Segmentations of defect-free gear blank casting image A-1.

(a) Ran

like cas

boundari

Initial Seg

As mentio

and Jain's

CLUSTER

of a featu

no more th

coordinate

six-dimens

image. Th

a 5×5 neig

clustering si

CLUSTE

process is rep

{1,2,...,K}

clusters. Set {

(a) Range Image of spindle-like casting.

(b) Flynn, Jain, and Hoffman Segmentation.

(c) New Segmentation.

Figure 5.4. Segmentations of defect-free spindle-like casting.

boundaries.

Initial Segmentation - CLUSTER

As mentioned above, the initial segmentation of the image was accomplished using Hoffman and Jain's technique. Their method used an unsupervised classification algorithm called CLUSTER [86]. CLUSTER minimizes a squared error function that measures the distance of a feature vector from cluster centroids. The input to CLUSTER was a sampling of no more than 1000 range pixels with six features drawn from each pixel. The x, y, and z coordinate of each pixel along with the corresponding surface normals formed CLUSTER's six-dimensional input. Figure 5.5 shows six features associated with each point in the range image. The surface normals were computed using a principal components analysis on a 5×5 neighborhood about each pixel. CLUSTER can be categorized as a local feature clustering since the features are local image properties [55].

CLUSTER performs classification in an iterative two-step process. The following process is repeated until none of the clusterings change during an iteration. First, a set of $\{1, 2, \ldots, K\}$, where K = 20, clusters is generated sequentially, where set $\{i\}$ contains i clusters. Set $\{i + 1\}$ is formed by selecting the pattern that is the greatest distance from the

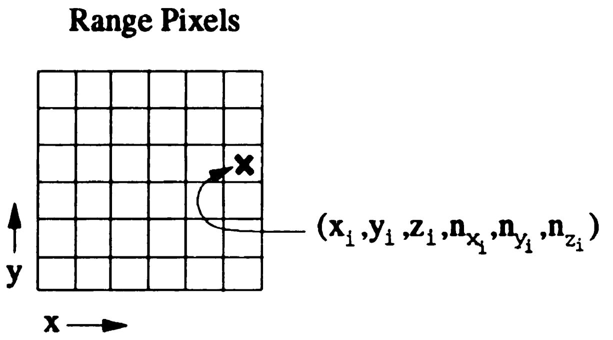


Figure 5.5. Clustering was performed using six features of each pixel P_i, including the coordinates of the point and the x, y, and z components of the normal vector $\vec{n_i}$ at the point.

set of $\{i\}$ cluster centers as the center of the new cluster. The total squared error for each of the K sets of clusterings is calculated and retained. Within each set of clusters, patterns are classified as members of the cluster whose center is the shortest distance away. The second step within an iteration creates a new set of clusterings using a greedy merging technique. In the greedy merging, if any two clusters within the ith clustering can be merged to produce a smaller total squared error than that of the previous $(i-1)$th clustering, then the clusters are merged and the old $(i-1)$th clustering is replaced with the new clustering.

After the iterations, the resultant sets of clusterings are examined to determine which is "best." A statistic $M(k)$, proposed by Coggins and Jain [38], measures the compactness and separation of the clusters. The clustering containing k clusters, where k is as close to K as possible and for which $M(k)$ is at a local maximum, is chosen as the correct clustering. Each pixel in the image is then labeled as a member of whichever cluster is closest, using a minimum Euclidean distance classifier to measure the distance of the pixel feature vector from the feature vector associated with the center of the cluster.

The initial segmentation by CLUSTER usually produces over-segmentations, but the subsequent processing merges adjacent regions of similar surface types.

F

fr

5.

In

sch

is s

segı

and

v

of th

imag

imag

gear

pipe f

quite

were s

image

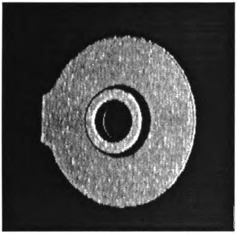


Figure 5.6. Range image of defect-free gear blank casting.

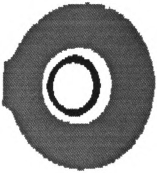


Figure 5.7. Segmentation of casting image shown in Figure 5.6.

5.1.3 Segmentation Results

In the remainder of this section, we discuss the performance of the revised segmentation scheme.

A pseudo-intensity rendering of a range image of a defect-free gear blank 356 casting is shown in Figure 5.6. Two planar surfaces are visible in this image. Our modified segmentation scheme correctly segmented the image into two regions (displayed as gray and black regions), as shown in Figure 5.7.

We applied the segmentation algorithm to 138 images of castings (25 different images of three instances of the 588x gear blank, 8 different images of the big cylinder, 9 different images of the cylinder, 16 different images of two of the spindle-like castings, 12 different images of two instances of the housing, and 68 different images of ten instances of the gear blank 356 castings) and to several non-casting images, including images of plastic pipe fittings, mannequin heads, and wooden blocks. The segmentation generally performed quite well, seldom creating heterogeneous regions. A total of 123 of the 138 casting images were segmented well. A characterization of the segmentation performance for the casting images is shown in Table 5.2. The table shows the number of good segmentations, over-

Identifier	Good Seg.	Over-seg.	Under-seg.	Poor Seg.
588x Gear Blank	24	0	0	0
Big Cyl	9	0	0	0
Cyl	8	1	0	0
Spindle	9	5	1	1
Housing	6	0	5	1
356 Gear Blank	67	1	0	0

Table 5.2. Characterization of segmentation performance.

segmentation, under-segmentations, and "poor" segmentations (segmentations that were over-segmented in one portion of the image and under-segmented in another portion of the image). Most of the problematic segmentations (the under-segmentations and "poor" segmentations) occurred in cylindrical regions, probably due to poor estimation of surface curvatures.

The scheme did have difficulties segmenting filleted regions on castings, however, and occasionally was overly agressive in the merging step. Filleted regions were often merged with one of the adjacent regions or else segmented into two or more patches representing sub-fillet regions. The merging step seemed to be too liberal for some images of cylindrical castings. This is especially troublesome because some of the cylindrical casting surfaces contain the remnant of a gate. The gate is also cylindrical and has been machined to a height of 0.05" above the surrounding surface. This is a very small difference that can probably be ignored during object recognition tasks that need only to distinguish between dissimilar objects. However, it is an important variance in the context of inspection and must be detected. If it is not detected by the original segmentation, then it must be detected by subsequent processing prior to inspection.

A pseudo-intensity rendering of the range image of a small cylindrical casting is shown in Figure 5.8. The segmentation of this image is shown in Figure 5.9. The segmentation algorithm did not merge any surface patches that should not have been grouped. Some surface patches were not identified by the segmentation scheme, however. For example,

a

I

s

a

(c

pi

de

in

alı

sm

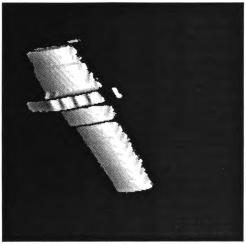


Figure 5.8. Pseudo-intensity rendering of defective cylindrical casting depth map.

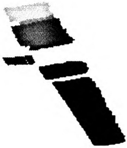


Figure 5.9. Segmentation of defective cylindrical casting of Figure 5.8.

the central ridged "ring" about the casting consisted of several very small patches that could not be merged together and which were too small to retain. A filleted region containing two small defects was also not retained by the segmentation. We address this problem below.

Several other segmentations of castings are shown in Figures 5.10 and 5.11. In these images, not all of the pixels in the images were labeled. This is because small regions that could not be merged with adjacent regions were removed during the segmentation. In Figure 5.11(b), the defect was segmented into the same region as the pixels which surrounded it. Therefore, the defect detection routines must be able to handle defects that are not isolated by the segmentation scheme. In Figure 5.11(c), the pit defect was too deep (concave) to return a depth value, so that region appears as a "void" (a connected region of pixels for which depth could not be sensed and which is contained within regions for which depth could be sensed) in both the range image and its segmentation. The excess material in the center of the casting was mostly merged with the small planar ring surrounding it, although a small third segment was also found at the junction of the center area and the small ring. Thus, presence of defects may cause slightly unpredictable segmentations. The u

i.

fo

se

ure

cal

def

was

Pro

One

pixels

those

the ne

feature

similar

†We a
regions to

defect detection schemes developed in this thesis can handle some false segmentations, although there are some limits. These issues are discussed in Chapters 7-9 on defect detection. In general, small defects are segmented into the same region as the surrounding points while large defects are either segmented into regions by themselves or segmented into several different regions. This implies that the inspection algorithms which operate on this data should search for small defects within larger regions and also search for regions that represent defects.

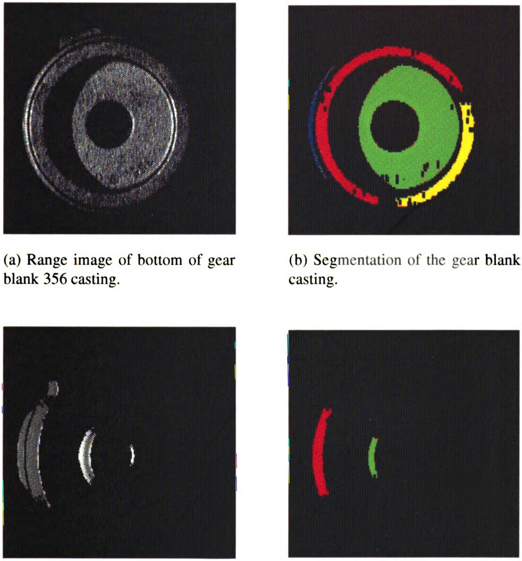
Segmentations of two images of a defect-free spindle-like casting are shown in Figure 5.12. The image in Figure 5.12(b) is over-segmented while the image in Figure 5.12(d) is incorrectly segmented. Segmentations of the latter type are particularly problematic for the defect detection schemes developed in this thesis, but only a few such incorrect segmentations have been observed in our experiments[†].

Large defects can also produce similar segmentation problems, as illustrated in Figure 5.13. The object in this image contains an excess material defect between two cylindrical regions. The defect in the image was segmented into two different regions. Part of the defect region was merged with the cylindrical region below it while the rest of the defect was merged with the filleted surface between the two cylinders.

Problems in Segmentation

One of the weaknesses of most range image segmentation schemes, including ours, is that pixels near jump boundaries are assigned no label (or are labeled incorrectly). The labels for those pixels cannot be recovered accurately because of their relative lack of context; that is, the neighborhood around the pixel is not homogeneous, making it difficult to extract stable features. Small regions, such as the ridged "rings" in Figure 5.8, also are not labeled for similar reasons. If the segmentation is only to be used by an object recognition module, this

[†] We are currently extending the segmentation scheme to use model information to allow over-segmented regions to be merged.



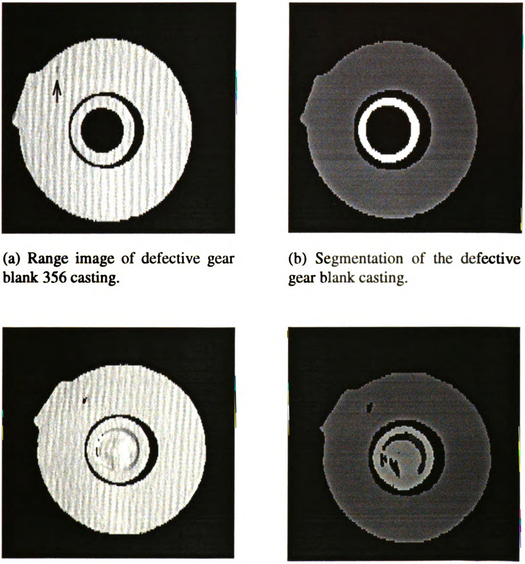
(a) Range image of bottom of gear blank 356 casting.

(b) Segmentation of the gear blank casting.

(c) Range image of side of gear blank casting.

(d) Segmentation of side of gear blank casting.

Figure 5.10. Range images and resulting segmentations of occluded castings.



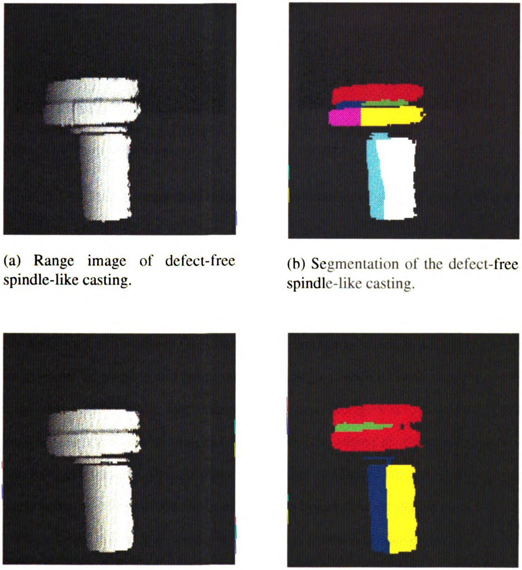
(a) Range image of defective gear blank 356 casting.

(b) Segmentation of the defective gear blank casting.

(c) Range image of gear blank casting with two defects.

(d) Segmentation of gear blank casting with two defects.

Figure 5.11. Range images and resulting segmentations of defective castings.



(a) Range image of defect-free spindle-like casting.

(b) Segmentation of the defect-free spindle-like casting.

(c) Range image of defect-free spindle-like casting.

(d) Segmentation of defect-free spindle-like casting.

Figure 5.12. Range images and resulting segmentations of spindle casting.

(a) F
the c

Figure
casting

is prob

allow

this de

range

makes

small

discus

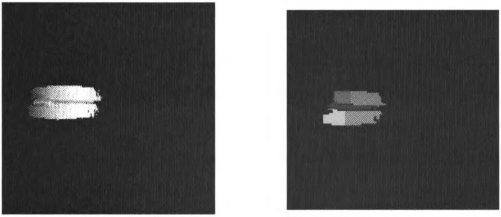
rarely

a defe

chapte

So

data w



(a) Pseudo-intensity rendering of the depth map of casting.

(b) Segmentation of the casting image.

Figure 5.13. Range image and resulting segmentation of part of a defective spindle-like casting.

is probably acceptable because the other extracted regions provide adequate information to allow successful matching to the stored object model. However, for inspection purposes this deficiency is a problem. It is desirable to inspect as much of the casting surface in the range image as possible and being unable to assign a label to pixels near region boundaries makes this more difficult. For example, in the case of Figure 5.8, the region containing the small amount of excessive material just “above” the central ring was not extracted. As was discussed in Chapter 1, defects commonly occur on filleted surfaces of the object, but only rarely occur on other curved surfaces. Thus, in Figure 5.8, the region most likely to contain a defect could not be inspected. We propose a solution to this problem at the end of this chapter.

Some of the segmentation problems are related to the deficiencies of the sensor. If range data were available in regions of self-occlusion, segmentation might be improved.

5.2

As s

base

requ

clas

clas

pixe

Besi

inte

it al

evid

loca

syst

desi

estin

tech

just

hypo

assu

gene

catio

exha

over

5.2 Surface Classification

As stated in Chapter 3, many 3D object recognition systems have used object models based on surface representations called *boundary representations* or *B-reps*. Using B-reps requires that the parameters of the sensed surfaces be estimated reliably. We treat surface classification as a separate step from segmentation and use the results of segmentation during classification. This is because the segmentation scheme did not use any surface model when pixels were grouped to form connected patches. Other techniques in the literature, such as Besl and Jain's [12] variable-order surface fitting procedure for segmentation, have used integrated segmentation and classification schemes.

Surface classification is important for object recognition and inspection systems because it allows the sensed surfaces to be correctly matched to model surfaces, thus providing evidence for the identity and pose of the object. In object inspection systems, object localization and/or identity may already be known, but it is still necessary for a vision system to compare the extracted surface shape properties of the object with the ideal and desired properties encoded in the object model.

Many papers in the literature have proposed surface classification and/or parameter estimation techniques (e.g. [23, 163]). Lozano-Peréz et al. [108] have presented a technique for finding the direction (but not the position) of cylinder axes in a scene using just three depth scans (two scans to hypothesize axis direction and the third to verify the hypothesis). They claimed high accuracy (up to one or two degrees), but their method assumed that there were actually cylinders present in the scene. They also suggested a general method for fitting cylinders and cones to range data that is similar to our techniques.

Recently, many researchers have used curvature estimates for quadric surface classification and parameter estimation. These methods are very appealing because curvatures are exhaustive descriptors of local surface shape and a collection of local curvature estimates over the surface can yield estimates of quadric surface parameters [56, 57]. In general,

curvature is estimated at each pixel using second derivatives of a local surface fit. Mean and Gaussian curvatures, or equivalently, the pair of principal curvatures, can then be used to classify each pixel. This strategy was explored by Flynn and Jain in [56] and the problems in curvature estimation were examined in [57].

The use of surface curvature has been popular in the computer vision literature. A discussion of curvature-based approaches to segmentation was presented in Section 5.1. As has been noted in that section, there are some difficulties in using curvature-based techniques. Most of the curvature-based segmentation techniques either implicitly or explicitly perform a region classification as part of the segmentation. A few additional efforts that focus more on curvature-based classification than on segmentation are summarized in this section. Yokoya and Levine [193], for example, have used principal curvatures to extract solids of revolution from range data. Hoffman and Jain [75] used curvature to classify surface patches as planar, convex, or concave. Bolle and Sabbah [15] used curvature information to classify synthetically generated planes, spheres, cylinders, and cones. Their method actually required third-order properties (i.e., change in curvature) to extract the parameters of conical surfaces, making it very susceptible to noise. Bhanu and Nuttall [13] used principal curvatures for surface characterization which was implemented on a shared-memory parallel processor.

Many researchers have used methods based on the Hough transform for surface parameter estimation. The Hough methods seem appropriate at times, especially if small amounts of noise or partial occlusion must be handled. Hough-based techniques have been very popular in 2D vision and have been occasionally useful in 3D vision, although simultaneous estimation of many parameters is demanding in both space and time. Yokoya and Levine [193] have used Hough techniques to determine the parameters of surfaces of revolution. Muller and Mohr [121] have presented a general framework for quadric surface parameter extraction using a divide-and-conquer Hough approach in the parameter space. Hebert and Ponce [71] have used Hough techniques for classifying surfaces as planar, cylindrical, or conical patches by clustering projected points along surface normals onto a Gaussian sphere. Taylor [168] has presented a Hough-based approach for fitting spheres, cylinders, and cones in a parameter space arranged as a k-tree to allow for quick evidence accumulation. Only zeroth and first derivative measures are used in the voting although curvature is used as a consistency check to reduce the number of false hypotheses. Wahl and Biland [180] have recognized 3D polyhedral objects in 2D images by decomposing Hough spaces into polyhedral primitives.

5.2.1 Motivation for Fitting Technique

We use the Hough transform for spherical and cylindrical parameter estimation. In each case, the size of an accumulator in the parameter space depends on the surface model parameters. Overlapped bins are used to reduce the effects of quantization.

Our suite of techniques relies only on first-derivative and positional (spatial) information for estimation of spherical, conical, and cylindrical surface parameters. We do not use second derivative quantities such as curvature which are unreliable in the presence of noise. Geometric characteristics of the known surface types are exploited to extract surface parameters. Model parameters (radii of cylinders and spheres and vertex angles of cones) are used to constrain the surface fitting. This reduces the number of free parameters in the surface fitting problem and improves the parameter estimates. We use the Hough-based techniques in the extraction of spherical and cylindrical parameters and least squares methods in estimation of conic parameters.

Since a large majority of man-made parts can be modeled exactly or at least well-approximated by a small set of planar or quadric surfaces [15, 111], our approach to surface fitting is appropriate.[‡] Among the quadrics, the natural quadrics (spheres, cylinders, and cones) are observed often in machined parts.

[‡] Marshall [111] has noted that 80-90% of manufactured parts that a typical industrial company (such as British Aerospace) needs to inspect consist solely of planar, cylindrical, and/or spherical faces.

Most of the available algorithms that fit quadric surfaces to the segmented surface patches use regression or some variant of regression. For planar surfaces, a simple linear regression procedure can be used to estimate the four parameters (a, b, c, d) of the planar surface as described by the implicit equation:

$$ax + by + cz + d = 0.$$

The planar surface parameters are not invariant to translation and rotation. Quadric surfaces can be characterized using the implicit quadric form

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0$$

which yields a linear regression procedure for fitting implicit quadrics to sampled data. Invariants of this implicit form have been used by Hall *et al.* [68] to identify surface type, but the behavior of the fitting in the presence of noise in the 3D data has not been explored. It may also be difficult to discriminate between different quadric forms based on estimated parameters.

Another difficulty in using the implicit quadric form is that known surface parameters cannot be easily injected into the fitting procedure. In model-based object recognition systems, it is usually possible to know which types of surfaces can be present in the data and the range of the surface parameter values. The fitting procedure that is used in this research uses information about surface type and known parameter values to estimate the remaining parameters.

After surfaces have been identified and their parameters estimated, an object recognition and/or inspection system can match image surfaces to model surfaces. This allows the identity and the pose of the object to be determined. In some inspection systems, it may not be necessary to explicitly determine object pose, however; the identity and pose of the

a

e

(w

tha

in t

nam

and

A

in Fi

the g

evalua

sensed

mean s

[1] The

input to th

paramete

to the clas

object may be known. In such cases, surface classification is useful as a precursor to surface shape inspection. For instance, it might be desirable to find how closely a curved surface in the data matches the parameters of the ideal surface encoded in the model.

5.2.2 Classification of Planar, Spherical, Cylindrical, and Conical Surfaces

This section presents the suite of techniques for planar, spherical, cylindrical, and conical surface classification in range imagery (additional details about these techniques are available in [126, 127]). The input data is assumed to be a set of 3D points and accompanying estimates of surface normals

$$\{(\mathbf{p}_1, \mathbf{n}_1), \ldots, (\mathbf{p}_n, \mathbf{n}_n)\}$$

(where $\mathbf{p}_i = (x_i, y_i, z_i)$ and $\mathbf{n}_i = (n_{x_i}, n_{y_i}, n_{z_i})$). The classification technique also assumes that "homogeneous" regions (i.e., smooth surface patches) have already been identified in the image by the segmentation scheme outlined in section 4.2.2[§] Model information, namely known surface types and their geometric parameters, constrains the fitting problem and serves as a guide to the extraction of the remaining unknown surface parameters.

A block diagram showing the processing in the surface classification algorithm is shown in Figure 5.14. First, the various model surfaces are fit to the sensed surface patches. Then, the goodness-of-fit between each transformed model surface and the sensed surfaces is evaluated. The model surface hypothesis that minimizes a uniform error measure between sensed and model data is accepted. Error in fit is measured using the average RMS (root mean squared) distance between sensed surface points and the hypothesized model surface.

[§] The segmentations produced by the Flynn- Hoffman-Jain segmentation were not always acceptable for input to the classification. Specifically, many cylindrical regions were too small to allow recovery of surface parameters. However, the new segmentation technique discarded fewer surface points, providing better input to the classification algorithm and improved parameter estimation.

This algorithm fits all model surfaces to each patch in the image, so the order in which surfaces are fit is not critical.

The errors resulting from fitting the best sphere, cylinder, cone, and plane to the surface patch are compared. The classification and parametrization that fits the patch with the smallest error is chosen as the best fitting surface. The error in fit is also used by the surface shape inspection method described in Chapter 8.

It is assumed that each patch in the image is an instance of one of the model surfaces. Thus, there is no reject option in this algorithm. However, a reject option could be added by use of a threshold on error in fit.

Some of our fitting techniques use linear least squares procedures, implying an assumption of Gaussian noise contaminating the range data. In experiments with a Technical Arts 100X White Scanner, this assumption seems very appropriate (Chapters 4 and 8 contain more information about the assumption of Gaussian noise in the scanner). If the range data were collected with some other sensor that had non-Gaussian noise, then robust regression techniques might be preferable to the traditional least squares methods used here. Furthermore, in such cases it might be desirable to use an error measure based on mean absolute deviations instead of the RMS fitting error used in our model-based fitting procedures.

Fitting Planes

The data-driven approach presented in [56] can be used to fit planes to surface patches. A plane is fit to each surface patch using the principal components of the set of points $\{\mathbf{p}_i, \mathrm{i} = 1, 2, \ldots, n\}$ on the surface. The plane that minimizes squared perpendicular error is selected as the best-fitting plane. The principal component analysis (PCA) technique [93] is well-known in exploratory data analysis [¶] It determines a vector $\hat{\mathbf{n}} = (\hat{n_x}, \hat{n_y}, \hat{n_z})$ as a surface normal estimate. Given $\hat{\mathbf{n}}$, the estimated maximum distance $\hat{d}$ between the plane

[¶] It is used to reduce the dimensionality of a data set or to extract new features from the original data that are not correlated. Our use of PCA in surface classification arises from a geometric interpretation of the vector-valued portion of the technique's output.

K
M
Sp
R

K
M
C
R

I
M
C
V
A

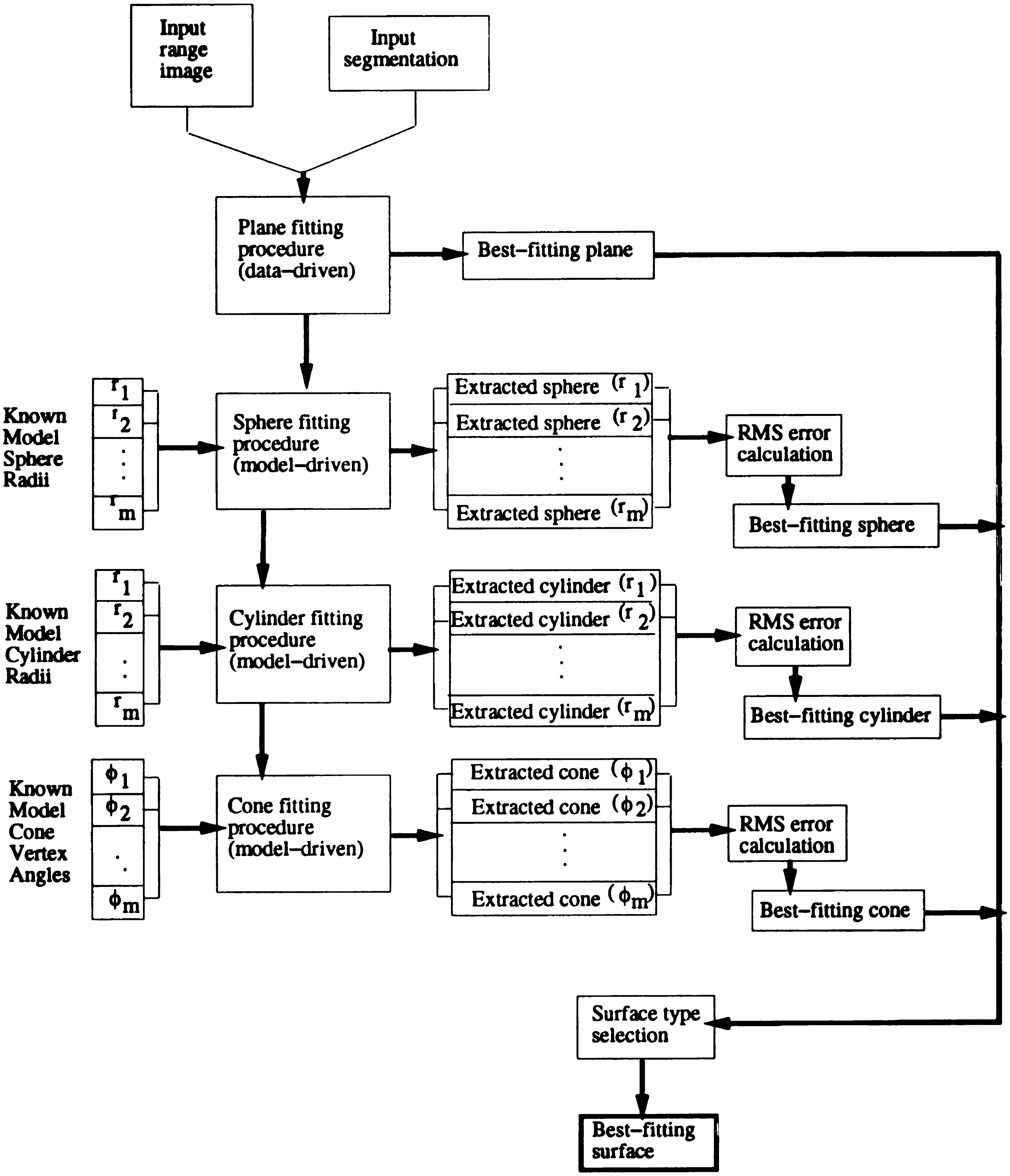

Input range image
Input segmentation
Plane fitting procedure (data–driven)
Best–fitting plane
Known Model Sphere Radii
r_1
r_2
r_m
Sphere fitting procedure (model–driven)
Extracted sphere (r_1)
Extracted sphere (r_2)
Extracted sphere (r_m)
RMS error calculation
Best–fitting sphere
Known Model Cylinder Radii
Cylinder fitting procedure (model–driven)
Extracted cylinder (r_1)
Extracted cylinder (r_2)
Extracted cylinder (r_m)
RMS error calculation
Best–fitting cylinder
Known Model Cone Vertex Angles
ϕ_1
ϕ_2
ϕ_m
Cone fitting procedure (model–driven)
Extracted cone (ϕ_1)
Extracted cone (ϕ_2)
Extracted cone (ϕ_m)
RMS error calculation
Best–fitting cone
Surface type selection
Best–fitting surface

Figure 5.14. Surface classification procedure block diagram.

I

A

ra

O

3D

is

sur

I

t

and the origin is

$$\hat{d} = -\hat{\mathbf{n}} \cdot \mathbf{m},$$

where $\mathbf{m}$ is the sample mean vector of the n points $\mathbf{p}_i$. The RMS fitting error for the plane fit to the patch is defined as

$$E = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[\hat{\mathbf{n}} \cdot \mathbf{p}_i + \hat{d}\right]^2} \ . \qquad (5.1)$$

Since the geometric parameters of planes are not invariant to rotation, model- driven planar fitting was not investigated. This plane-fitting procedure requires $O(n)$ time and $O(n)$ space and takes less than a second per segment on a Sun SPARCstation 2.

Fitting Spheres

A sphere can be characterized by its radius r and the position of its center (x_0, y_0, z_0). The radius r is invariant under rotation and translation and is known prior to the fitting process. Our sphere-fitting method accumulates evidence for the location of the sphere center in a 3D parameter space (a Hough space). For each model radius r_j, $j = 1, 2, ..., M$, where M is the number of spherical models, we estimate the best centroid $\hat{\mathbf{c}_j} = (\hat{c_{x_j}}, \hat{c_{y_j}}, \hat{c_{z_j}})$ of the surface in the range image. The following procedure is used for each radius r_j.

1. An empty three-dimensional accumulator array is created. This array has dynamically-sized bins whose size varies with the radius r_j. The bins are overlapping and of size $r_j/4$. The overlap reduces the effects of quantization noise. Each cell is logically overlapped by 25% of the bin size in each dimension. During the voting procedure, if any vote falls within the upper 25% of the range of any dimension of the bin, both that bin and the appropriate neighboring bin(s) are incremented. A two-dimensional analog of the overlapping bins is shown in Figure 5.15.

The array encloses a subset of $\mathbf{R}^3$ given by

$$[x_{min} - r_j, x_{min} + \frac{18}{5}r_j] \times [y_{min} - r_j, y_{min} + \frac{18}{5}r_j] \times [z_{min} - r_j, z_{min} + \frac{18}{5}r_j],$$

where x_{min} , y_{min}, and z_{min} are the minimum x, y, and z coordinate values, respectively, in the region. Similarly, we define x_{max}, y_{max}, z_{max} to be the maximum x, y, and z coordinate values in the region to be fit. This subspace is guaranteed to contain the center of any sphere of radius r_j passing through the sampled points $\mathbf{p}_i$. The accumulator array has 23 quantiles along each of three dimensions (for a total of 23^3 or $12,167$ bins) ‖ The accumulator array should have a minimum of 20 cells along each dimension, although a total of 23 cells were used in this work to allow for some noise and quantization error to be present in the sensed data. Larger voting spaces improved the fitting little while smaller spaces resulted in an increased number of poor fits.

2. Each surface point $\mathbf{p}_i$ and its associated normal vector estimate $\hat{\mathbf{n}}_i$ yields two estimates $\mathbf{p}_i \pm r_j\mathbf{n}_i$ of the sphere's center location for radius r_j, corresponding to a concave or convex surface, respectively. These two position estimates cause two or more Hough array accumulators to be incremented. Each vote in the accumulator array has as its attributes the estimated coordinates of the center.

3. Following the processing of all points $\mathbf{p}_i$, the cell of the Hough space that contains the most votes is determined. The attributes of the votes contributing to that cell are averaged to produce a location estimate $\hat{\mathbf{c}}_j$ for the hypothesized sphere of radius r_j.

This Hough procedure is performed for each model radius r_j, producing M center estimates $\hat{\mathbf{c}}_j$. These hypothesized spheres are then evaluated using the sample surface

‖Due to the overlapping bins, the size of the Hough space is $\frac{23}{5}r_j$ in the x, y, and z directions.

s

o

ra

Fi

A r

axis

assu

colle

as fo

l.

points $\mathbf{p}_i$ and a standard RMS error measure:

$$E_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2} \ , \tag{5.2}$$

where $d_i = \sqrt{(x_i - \hat{c_{x_j}})^2 + (y_i - \hat{c_{y_j}})^2 + (z_i - \hat{c_{z_j}})^2} - r_j \ , \ \ j = 1, 2, ..., M$. The hypothesized sphere with the lowest E_j is labeled the most likely spherical parametrization of the surface patch.

The time complexity of this fitting procedure is $O(nM)$, where n is the number of input points and M is the number of possible model radii. The space complexity is $O(n)$, due to the storage needed for the set of input points $\mathbf{p}_i$. The 3D accumulator array consumes a constant 0.2 megabyte in the current implementation. Fitting takes no more than a few seconds per patch (on a SPARCstation 2) per model radius. (The specific time depends on the size of the patch. We observed times between 0.21 and 1.56 seconds in 240 × 240 range images.)

Fitting Cylinders

A right circular cylinder can be uniquely defined using three parameters: a point P_0 on the axis, the axis direction $\vec{a}$, and the radius r. Similar to the fitting method for spheres, it is assumed that all possible radii of model cylinders are known *a priori* and drawn from a collection of M possible values $\{r_1, r_2, ..., r_M\}$. Our earlier fitting strategy proceeded [127] as follows.

1. An empty accumulator array A was built. The entries accumulated votes for points on the axis $\vec{a}$. The array contains 23 quantiles along each axis with a bin size of $r_j/4$ and 25% overlap. It again corresponds to the following region in $\mathbf{R}^3$

$$[x_{min} - r_j, x_{min} + 3.6r_j] \times [y_{min} - r_j, y_{min} + 3.6r_j] \times [z_{min} - r_j, z_{min} + 3.6r_j].$$

2. Possible axis points on the cylinder are updated by incrementing the accumulators corresponding to $\mathbf{p}_i \pm r_j\mathbf{n}_i, i = 1, 2, ..., n$. This is analagous to the accumulation of estimates for the center of spheres as described above, except that the points accumulated will be distributed along the hypothesized axis, not concentrated at a specific location in the accumulator array. Each point $\mathbf{p}_i$ increments (at least) two cells allowing the extraction of parameters for both concave (e.g., troughs) and convex cylinders. Votes that fall outside the accumulation region are discarded.

3. The hypothesized axis $\vec{a}$ is extracted from the accumulator array by performing a principal components analysis [86] on the set H of 3D points whose elements are the centroids of the projected points that fall within high-count cells. High-count cells are defined to include the highest-population bin, b_{max}, and all other bins with 50% or more of the maximum population, $\mathrm{count}(\mathrm{b}_{max})$,. Details of the principal component technique are available in [93] and Dubes. The analysis produces an estimate of the 3D direction along which the variance of the data is largest; this direction is taken as a hypothesized direction of the cylinder's axis $\vec{a}$. The centroid of the highest-count cell is hypothesized as the axis point P_0.

This original technique was found to be unsuitable for inspection purposes, however. In some cases, the fitting exhibited what would appear to be acceptable RMS error measures, but upon examination of the local deviations of the data from the surface, some portions of the surfaces were found to have unacceptably large deviations. By "unacceptable," we mean that the deviations were larger than the tolerances for insufficient material (0.06"). It was also found that the fitting was very poor (or not possible) for many cylinders whose axis length was much smaller than their radius. A final cause for concern was that the depth data near some edges in the image (specifically, points near certain limb edges where the points' normals had a large horizontal component) appeared to be noisier than data from near the "top" of a cylindrical region.

s

$=$

T

er

nc

co

$=$

$-\frac{6}{5}$

a pc

(

the I

heigh

type t

best a

An attempt was made to resolve the latter problem by restricting the surface points that participated in the Hough voting technique. This was achieved by applying a morphological constriction operation to the surface segment. However, this technique actually produced poorer fittings. Constrictions of 1, 2, and 3 pixels were tried, and each produced increasingly poorer fittings. This problem is caused by a limitation of the sensor rather than the surface fitting procedure, so our proposed solution is to either use multiple scans or to restrict the orientation of the object. (Further details about this problem are presented in Section 5.2.3.)

The fittings were significantly improved by three enhancements to the original cylindrical fitting, though.

(1) The position of the Hough accumulator array A was improved using the following heuristic. The morphological constriction operation is repeatedly applied to the candidate region until the region is shrunk to a single pixel (or to a collection of curves of width one). This point (or, in the case of the collection of curves, the point that is first in a raster scan search of the image), called $\mathbf{p}_c$, is used as a cue for the location of the Hough space. If $\mathbf{p}_c = (x_c, y_c, z_c)$, then the origin of the Hough space is set to $(x_c - \frac{3}{2}r_j, y_c - \frac{3}{2}r_j, z_c - \frac{3}{2}r_j)$. This change produced some improvements in fitting because it allows the Hough space to enclose more votes from the central part of the region. If this positioning technique does not yield at least 3 high-count cells to "vote" for the axis, a new position for the space is computed and then voting is reconducted. The new position uses a different seed point $\mathbf{p}_c = \frac{1}{2}(x_{min} + x_{max}, y_{min} + y_{max}, z_{min} + z_{max})$. The origin of the Hough space is set to $\mathbf{p}_c - \frac{6}{5}r_j(1, 1, 1)$. Originally, a coefficient of $\frac{3}{2}$ was used instead of $\frac{6}{5}$, but it was found to yield a poorer fit than the final coefficient.

(2) We also refined our original definition of what constituted a high-count cell within the Hough space. This was especially effective in improving the fit for cylinders whose height was smaller than their radii. Under the original fitting technique, cylinders of this type typically had less than four high-count cells, making parameter estimation unstable at best and impossible at worst. By relaxing the definition of a high count cell, the collection

(

E

(.

cy

th

the

wi

whe

assoc

**W

very li

[††]Sn

smaller

B_c of high-count cells was enlarged. All cells whose count was at least $\frac{\text{count}(b_{max})}{3}$, where $\text{count}(b_{max})$ is the count in the highest-count cell, were used to estimate axis direction using principal component analysis. For small cylinders, this usually added one or two cells to the set B_c. The position of the axis point P_0 was also estimated differently. P_0 was hypothesized as the weighted average of the elements of B_c. The population of each high-count cell was used as the weight in this computation.

(3) A multi-resolution, or foveal [22], approach to fitting was adopted for cylinders whose height was less than their radius r_j.[**] For these cylinders, the above fitting procedure, including the new enhancements, was followed, although the axis position and orientation estimation steps were not immediately carried out. Instead, the Hough space was refined and a second accumulation was conducted. The centroid, $\mathbf{p}_m = (x_m, y_m, z_m)$, of the highest count bin in the first accumulation was used as a focus point (i.e., the approximate center) for the new space. In the new space, a bin size of $\lceil \frac{h_j}{4} \rceil$ was used [††], where h_j is the height of cylinder j with radius r_j. The origin of the new Hough space was approximately $(x_m, y_m, z_m) - \frac{21}{2} \lceil \frac{h_j}{5} \rceil (1, 1, 1)$. This refinement technique greatly improved fitting for small cylinders.

This procedure is invoked for each model radius, r_j. The hypothesized cylinders are then evaluated. As with spheres, an RMS error of fit is computed and the radius yielding the smallest error is accepted as the most likely cylinder. The RMS error in fit associated with the cylinder of radius r_j is

$$E_m = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left\{ \left| \, \| (p_i - P_0) \times \vec{a}_j \| - r_j \right|^2 \right\}} \ , \tag{5.3}$$

where p_i is a point on the surface patch, P_0 is the axial point, $\vec{a}_j$ is the axis direction associated with r_j, $j = 1, 2, ..., M$.

[**] We initially attempted foveation for cylinders whose height was less than $2r_j$, but this improved fitting very little.

[††] Smaller bin sizes tended to produce poorer fits because they yielded cells that were too small and a smaller space that discarded too many votes.

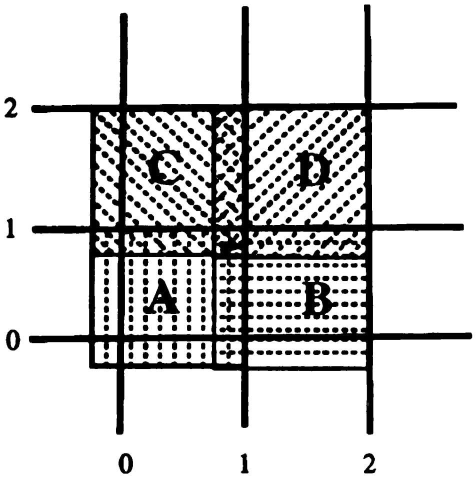


Figure 5.15. An example showing overlapping bins for Hough Transform. The data point at (0.8,0.8) contributes votes to bins A, B, C, and D.

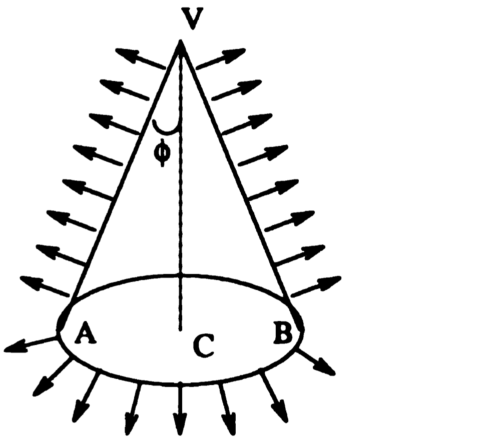


Figure 5.16. Sample cone showing axis, vertex angle, and normal vectors along limb edges.

As with the sphere-fitting procedure, time complexity for this fitting method is linear in the number M of hypothesized radii r_j and the number n of input points $\{\mathbf{p}_i\}$. Storage requirements are linear in the number of points with an additional amount required for the parameter array. Each run of the program takes no more than a few seconds per model radius and per surface segment. Execution times are slightly larger than the sphere-fitting technique.

Fitting Cones

The strategy to fit cones to sets of 3D points differs from the cylinder- and sphere-fitting techniques. Cones are represented by three parameters, the axis $\vec{a}$, the vertex $v = (x_0, y_0, z_0)$, and the vertex angle ϕ between the axis and surface. First, the axis $\vec{a}$ is estimated. Then the estimate of $\vec{a}$ is used to estimate the vertex v. The vertex angle ϕ can be selected from a pre-specified set $\{\phi_1, \phi_2, ..., \phi_n\}$ or estimated. This yields both model-driven and data-driven approaches to fitting.

Figure 5.16 shows a sample cone including normal vectors, the axis $\overrightarrow{\mathrm{VC}}$, and two vectors

$\overrightarrow{\mathrm{VA}}$ and $\overrightarrow{\mathrm{VB}}$, which are surface rulings. The vertex angle of the cone is ϕ. The intersection of all the planes tangent to the surface of the cone is the vertex v of the cone. This characteristic is the basis for the technique to estimate v.

Each surface point $p_i = (x_i, y_i, z_i)$ and its associated normal $\vec{n_i} = (n_{x_i}, n_{y_i}, n_{z_i})$ defines a plane tangent to the surface at p_i whose implicit equation is:

$$n_{x_i}(x_0 - x_i) + n_{y_i}(y_0 - y_i) + n_{z_i}(z_0 - z_i) = 0,$$

Hence,

$$n_{x_i}x_0 + n_{y_i}y_0 + n_{z_i}z_0 = n_{x_i}x_i + n_{y_i}y_i + n_{z_i}z_i.$$

Therefore, the following system of linear equations is solved for the vertex position $v = (v_x, v_y, v_z)$:

$$\begin{bmatrix} n_{x1} & n_{y_1} & n_{z1} \\ n_{x2} & n_{y_2} & n_{z2} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ n_{xn} & n_{y_n} & n_{zn} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} n_{x1}x_1 + n_{y_1}y_1 + n_{z1}z_1 \\ n_{x2}x_2 + n_{y_2}y_2 + n_{z2}z_2 \\ \cdot \\ \cdot \\ \cdot \\ n_{xm}x_n + n_{y_n}y_n + n_{zn}z_n \end{bmatrix}$$

Once the vertex v has been found, it is possible to determine the axis and vertex angle ϕ of the cone. Since the scalar product of a unit vector from the vertex to any point on the conical surface with the unit vector in the direction of the conic axis is constant, the axis direction $\vec{a}$ and vertex angle ϕ can be determined by a least squares method. A unit length vector expressing axis direction $\vec{a} = (a_x, a_y, a_z)$ is estimated using $\frac{\overrightarrow{vp_i}}{||\overrightarrow{vp_i}||} \cdot k\vec{a} = k\vec{a}\cos\phi$, where $\frac{\overrightarrow{vp_i}}{||\overrightarrow{vp_i}||}$ is the unit vector in the direction from the vertex v to a point p_i and $\vec{a}$ is a vector in the direction of the axis, $\overrightarrow{vC}$. Therefore, we solve

$$\begin{bmatrix} c_1(x_1 - x_0) & c_1(y_1 - y_0) & c_1(z_n - z_0) \\ c_2(x_2 - x_0) & c_2(y_2 - y_0) & c_2(z_n - z_0) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ c_n(x_n - x_0) & c_n(y_n - y_0) & c_n(z_n - z_0) \end{bmatrix} \begin{bmatrix} ka_x \\ ka_y \\ ka_z \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$

for $k\vec{a}$, where c_i is $\frac{1}{\|\mathrm{p}_i - v\|}$.

In this formulation, $k\vec{a}$ is the vector $\vec{a}$ scaled by $(1/\cos\phi)$. Hence, the length of $k\vec{a}$ is $(1/\cos\phi)$. Thus, estimates of both the axis $\vec{a}$ and vertex angle ϕ could be extracted from the output of this regression procedure. However, it is assumed that n distinct vertex angles $\{\phi_1, \phi_2, ..., \phi_n\}$ are present in conical surfaces in the object model database. This *a priori* information is used to verify estimates $\vec{a}$ and $\hat{v}$ as well as to select the appropriate ϕ, as follows: for each known vertex angle ϕ_j, the squared error-of-fit between the points p_i and the hypothesized cone is calculated and the ϕ_j yielding the minimum error is accepted.

The RMS fitting error for the hypothesized cone with vertex angle ϕ_j is defined as

$$E_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[\|\overrightarrow{vp_i}\| \ \sin\left|\cos^{-1}\left(\frac{\overrightarrow{vp_i}}{\|\overrightarrow{vp_i}\|}\vec{a}\right) - \phi_j\right|\right]^2} \ . \qquad (5.4)$$

The technique to estimate conic parameters can be used in a purely data-driven fashion by accepting the estimate of ϕ implicitly resulting from the least square estimate of axis orientation. Since the length of $k\vec{a}$ is $(1/\cos\phi)$, ϕ can be estimated using

$$\phi = \arccos\frac{1}{\| k\vec{a} \|} \ . \qquad (5.5)$$

Model-driven fitting is performed in real scenes, however, to ensure that a vertex angle present in the pre-specified set is selected. This allows instances of known conical surfaces in the model database to be detected. We accept the cone that fits the data with the smallest error. The model-driven cone fitting procedure takes at most a few seconds per segment per model vertex angle. We observed CPU times between 0.18 and 1.75 seconds in 240 × 240 range images on a SPARCstation 2. The time complexity is linear in both M and n. The data-driven procedure's time requirement is linear in n, since no *a priori* parameters are used. Space requirements are linear in the number of points n and radii M.

Selecting the Best Surface

In the preceding sections, specialized fitting procedures were presented for spheres, cylinders, cones, and planes. The quality of each fit was measured by the RMS perpendicular distance between the sampled data points and the fitted surfaces. Since these error measures are identical in character, their values can be compared directly to determine the best-fitting surface among the four types (i.e., the surface type with the smallest RMS error). Currently, it is assumed that each patch in the image is an instance of one of the model surfaces. Hence, there is no reject option in the algorithm. A reject option could be included by using an error threshold. Setting the appropriate threshold, however, is not trivial.

5.2.3 Surface Fitting Results

The surface fitting algorithm has been tested using synthetic and real range data of 3D objects and surfaces. Synthetic data was used to study the performance degradation under increasing levels of additive Gaussian noise. Experiments with real range data collected with the Technical Arts 100X range sensor demonstrate the techniques' usefulness for object recognition and inspection.

Radius	$\hat{x}$	$\hat{y}$	$\hat{z}$	Error
0.90"	-0.0006	-0.0006	0.0752	0.055
1.00"	0.0002	0.0002	-0.0004	0.0003
1.05"	-0.0001	-0.0001	-0.0333	0.029
1.15"	0.0084	0.0084	-0.1043	0.081

Table 5.3. Noise-free synthetic spherical data tested against spherical models. True radius of the sphere is 1.00".

Synthetic Image Data

Both noise-free and noisy depth data were generated to test the performance of the fitting technique. Noise-free depth data from the surfaces of spheres of radius 1", cylinders of radius 1", and cones of 45° vertex angles were generated. Figure 5.17 shows plots of the noise-free synthetic surfaces. Spheres, cones, and cylinders corrupted by various levels of additive zero-mean Gaussian noise were also generated. A cylindrical surface corrupted by Gaussian noise added to the z-coordinate is shown in Figure 5.18. For all test cases, data points were sampled on a 6.4" × 6.4" grid with 240 rows and 240 columns. For the experiments on noisy synthetic data, 1000 Monte Carlo trials were performed for each surface type to estimate the classifiers' performance.

Table 5.3 summarizes fitting results for noise-free synthetic spherical data of radius 1" centered at the origin against various model radii. The noise-free data is drawn from the positive-z hemisphere of the unit-radius sphere centered at the origin. Since the minimum error is associated with the case of r=1.0 (the true radius), the hypothesis that the sphere is of unit radius with center location $\hat{\mathbf{c}} = (\hat{x}, \hat{y}, \hat{z}) = (0.0002, 0.0002, -0.0004)$ is accepted. In Table 5.3, results are listed only for model radii close to the true radius. Spheres of other radii were also fit to the data, but the resulting error was quite large (therefore, those incorrect radii would also be rejected).

Table 5.4 contains the results of testing spherical data with two different levels of random noise against the model spheres of different radii. The results for both of these experiments

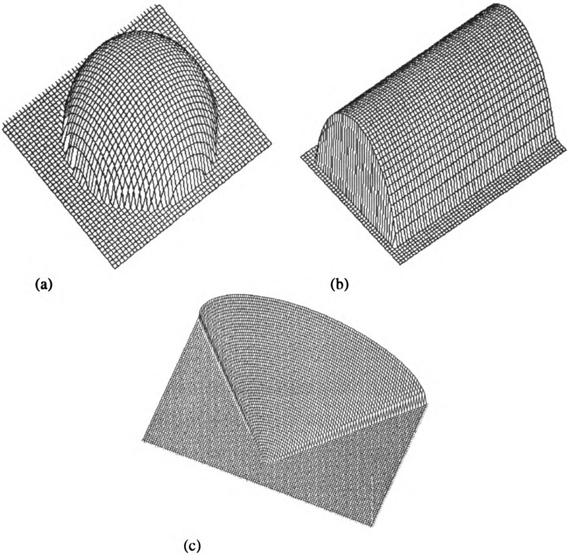


Figure 5.17. Synthetic primitive surfaces: (a) sphere, (b) cylinder, (c) cone.

	$\sigma = 0.01$				$\sigma = 0.1$			
r	$\hat{x}$	$\hat{y}$	$\hat{z}$	Error	$\hat{x}$	$\hat{y}$	$\hat{z}$	Error
0.90"	0.0022	-0.0021	0.0823	0.052	-0.0123	-0.0120	0.2354	0.099
1.00"	0.0080	0.0080	0.0142	0.013	0.0459	0.0455	0.1343	0.116
1.05"	-0.0080	-0.0080	-0.0310	0.031	-0.0579	-0.0577	0.1234	0.150
1.15"	0.0173	0.0173	-0.0979	0.086	0.0296	0.0375	0.1453	0.258

Table 5.4. Spheres fit to synthetically generated spherical data of radius 1.0 centered at the origin. Noise is sampled from a Gaussian distribution with $\sigma = 0.01$ and $\sigma = 0.1$ and was added to the true z coordinate of the spherical surface data. Parameter estimates and errors were averaged over 1000 Monte Carlo trials.

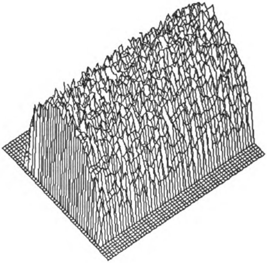


Figure 5.18. Noisy synthetic cylinder of radius 1.0" with additive Gaussian noise of standard deviation σ=0.05 present in the z-coordinate.

are averages computed from 1000 Monte Carlo trials. As expected, RMS fitting error increases with higher levels of noise. Under high levels of noise (e.g., $\sigma = 0.1$), an incorrect spherical hypothesis could be accepted. For example, the mean error for a sphere of radius 0.9" was lower than the mean error resulting in fitting spheres of radius 1.0". However, this level of noise ($\sigma = 0.1$) is not realistic for the data acquired with the Technical Arts 100X sensor; carefully collected range data should not be corrupted to the extent present in this example. Even if a parameter estimation error of this type did occur, an object recognition and/or inspection system could resolve the problem by accepting multiple spherical hypotheses that yielded nearly identical error measures. Naturally, the inspection system could falsely reject an object if the fit parameters varied too much from the expected model parameters. Thus, it is important that measurement error resulting from sensor noise be less than the allowable dimensional measurement tolerances in the inspection task.

The cylinder-fitting technique was tested with synthetic data extracted from a cylinder of unit radius with axis direction $\mathbf{a} = (a_x, a_y, a_z) = (1, 0, 0)$ and passing through the

	Noise-free							$\sigma = 0.01$			
r	$\hat{p}_x$	$\hat{p}_y$	$\hat{p}_z$	$\hat{a}_x$	$\hat{a}_y$	$\hat{a}_z$	Error	$\hat{p}_x$	$\hat{p}_y$	$\hat{v}_z$	Error
0.90	-2.77	0.00	0.08	1.00	0.00	0.00	0.04	0.05	0.00	0.09	0.038
1.00	-2.93	0.00	0.00	1.00	0.00	0.00	0.00	-0.07	0.01	0.01	0.013
1.05	-2.92	0.00	-0.04	1.00	0.00	0.00	0.02	-0.07	-0.01	-0.03	0.028
1.15	-2.88	0.00	-0.13	1.00	0.00	0.00	0.05	-0.06	0.03	-0.11	0.066

Table 5.5. Fitting cylindrical models to noise-free and to noisy synthetic cylindrical data with $r = 1.0''$, $\mathbf{a} = (a_x, a_y, a_z) = (1.0, 0.0, 0.0)$, and axis containing the origin. The synthetic data was corrupted by additive Gaussian noise with standard deviation $\sigma = 0.01$. $(\hat{p_x}, \hat{p}_y, \hat{p}_z)$ is the average estimated location of a point on the axis of the cylinder and $(\hat{a_x}, \hat{a}_y, \hat{a}_z)$ is the hypothesized axis direction (which was essentially identical in both tests). Parameter estimates and error for noisy data averaged over 1000 Monte Carlo trials.

origin. Table 5.5 contains the estimated parameters for the noise-free synthetic cylinder and for samples from cylinders corrupted by additive noise randomly sampled from a zero-mean Gaussian distribution with standard deviation $\sigma = 0.01$. The error columns in the table represent RMS error between data points and the hypothesized surfaces. Estimated parameters and RMS errors are averaged over 1000 Monte Carlo trials. For both the noise-free and noisy data, the smallest fitting error corresponds to the correct cylindrical hypothesis. However, as expected, the noisy data is not fit as well as the noise-free data.

The cone fitting technique was tested with synthetic data extracted from a cone with axis vector $\mathbf{a} = (a_x, a_y, a_z) = (1.0, 0.0, 0.0)$, vertex angle $\phi = \pi/4$ (about 0.785 radians), and with vertex location $\mathbf{v}$ at the origin. The results of model-driven fitting to noise-free and noisy data are shown in Table 5.6. For noisy data, 1000 Monte Carlo trials were performed and mean results compiled. The table lists the hypothesized vertex location, axis, and vertex angle for the data and the resultant error. Error is estimated using Eq. (5.4). A data-driven fitting to estimate ϕ was also explored. For the synthetic conic data, the data-driven fitting had a smaller mean error (approximately 0.01 for the three noise levels in Table 5.6) than the model-driven fitting, but it estimated a vertex angle slightly different from the true parameter $\phi = 0.785$ radians. Tests with Gaussian noise with $\sigma = 0.05$ and higher did not produce good fits.

σ	ϕ	$\hat{v_x}$	$\hat{v_y}$	$\hat{v_z}$	$\hat{a_x}$	$\hat{a_y}$	$\hat{a_z}$	Mean Error
0.0	0.785	0.0236	0.0000	0.0242	1.00	0.00	-0.03	0.0040
0.001	0.785	0.0237	0.0000	0.0242	1.00	0.00	-0.03	0.0043
0.01	0.785	0.0363	0.0000	0.0172	1.00	0.00	0.00	0.0210

Table 5.6. Fitting conical models to samples from synthetic cones with vertex $\mathbf{v} = (v_x, v_y, v_z) = (0.0, 0.0, 0.0)$, $\mathbf{a} = (a_x, a_y, a_z) = (1.0, 0.0, 0.0)$, and $\phi = 0.785$ radians. Parameter estimates (averaged over 1000 Monte Carlo trials) are reported for noise-free synthetic data and data corrupted by Gaussian noise with standard deviations $\sigma = 0.001$ and $\sigma = 0.01$. $(\hat{a_x}, \hat{a_y}, \hat{a_z})$ denotes the mean of the estimates of **a** and $(\hat{v_x}, \hat{v_y}, \hat{v_z})$ is the mean of the estimates of **v**.

σ	r	$\hat{x}$	$\hat{y}$	$\hat{z}$	Mean Error
0.0	0.9"	-1.173	-0.001	0.083	0.969
	1.0"	-1.333	0.000	0.003	1.020
	1.1"	-0.640	0.000	-0.077	0.984
	1.15"	-1.280	0.002	-0.130	0.911
0.1	0.90"	-0.022	0.050	0.611	0.483
	1.00"	-0.052	-0.008	0.330	0.608
	1.10"	-0.093	-0.007	0.495	0.560
	1.15"	-0.077	0.031	0.567	0.491

Table 5.7. Fitting spherical models to cylindrical data. Parameter estimates and error for noisy data have been averaged over 1000 Monte Carlo trials of synthetic cylindrical surfaces with $r = 1.0$" corrupted with additive Gaussian noise with $\sigma = 0.1$. $(\hat{x}, \hat{y}, \hat{z})$ is the mean estimated center position.

The above experiments demonstrate a low surface classification error rate for synthetic data. In Table 5.7 error rates are shown for fitting spheres to synthetic cylindrical data. The synthetic data was created in the same manner as the earlier tests. In comparing Table 5.7 to Table 5.5 we note that the cylindrical models fit the cylindrical data much better than any of the spherical models. The conical models and planes also did not fit the cylindrical data as well as cylindrical models. Classifications of other synthetic surfaces indicate no misclassifications of noise-free data and very few misclassifications of noisy data at the noise levels tested (uniform and Gaussian noise at levels up to $\sigma = 0.1$ for spherical and cylindrical surfaces and at levels up to $\sigma = 0.01$ for conical surfaces).

Figu
dept

Real

The

rang

Fig

th

cl

s

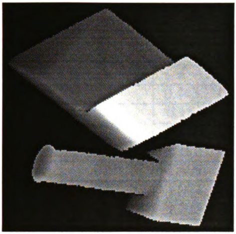


Figure 5.19. Intensity coding of depth.

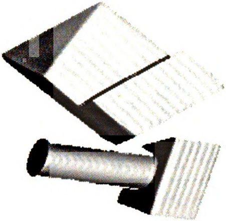


Figure 5.20. Pseudo-intensity rendering of range image.

Real Images

The performance of the surface classification techniques has been tested on a number of range images of isolated objects as well as to jumbles of occluded objects.

Table 5.8 shows the performance of the fitting techniques on the range image shown in Figure 5.19. The table shows fitting for the cylinder of radius 0.66" (the cylinder attached to the block). Figure 5.19 shows the range data rendered as intensity (lighter shaded points are closer to the sensor) and Figure 5.20 shows the "pseudo-intensity" image. The best-fitting surface is the correct surface, a cylinder of radius 0.66". Other cylindrical surface models exhibited larger RMS errors and other surface types fit the data even more poorly.

Table 5.9 shows the results of classification of a cylindrical pipe of radius 0.46" that was imaged in four different positions and orientations, one of which is shown in Figure 5.21. The image is displayed as an intensity-coded depth map. The results are compared with a data-driven parameter extraction method [56] by comparing the RMS error measures. In three of the four cases, the model-driven fitting performed better than the data-driven techniques. In the fourth case, the performance of the two techniques was essentially the

Surface
Cylinde
Spher
Cor

Table

An '*

Surface	Radius or ϕ	$\hat{x}$	$\hat{y}$	$\hat{z}$	$\hat{a_x}$	$\hat{a_y}$	$\hat{a_z}$	Error
Cylinder	0.5"	0.443	0.455	-1.728	(0.921,	-0.245,	-0.303)	0.0278
	0.6"	-0.183	0.639	-1.624	(0.930,	-0.206,	-0.303)	0.0113
	0.66"	-0.142	0.630	-1.694	(0.930,	-0.208,	-0.300)	0.0036
	0.7"	0.211	0.549	-1.849	(0.929,	-0.210,	-0.304)	0.0054
Sphere	0.5"	-1.409	0.886	-1.120	*	*	*	1.059
	0.6"	-1.004	0.831	-1.356	*	*	*	0.694
	1.1"	-1.225	0.818	-1.792	*	*	*	0.542
Cone	1.28	0.141	0.556	-1.038	(-0.352,	0.092,	-0.931)	0.104

Table 5.8. Data from the cylindrical surface in Figure 5.20 tested against various hypotheses. An '*' denotes that the specific parameter is not applicable for the surface type.

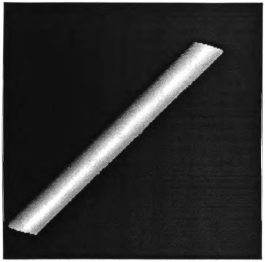


Figure 5.21. Intensity-coded range image of a pipe of radius 0.46" (Pipe 1).

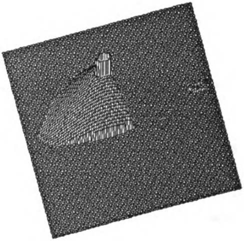


Figure 5.22. Profile plot of range image of a funnel.

Image
Pipe 1
Pipe 2
Pipe 3
Pipe 4

Table 5

Compa

extrac

same

0.46

the

par

fu

in

d

l

Image	Method	Surface	r	$\hat{p_x}$	$\hat{p_y}$	$\hat{p_z}$	$\hat{a_x}$	$\hat{a_y}$	$\hat{a_z}$	Error
Pipe 1	M	Cyl.	0.46	-2.02	0.41	-3.20	-0.480	-0.877	0.002	0.012
	D	Cyl.	0.373	1.44	6.74	-3.12	-0.467	-0.884	0.001	0.035
Pipe 2	M	Cyl.	0.46	-1.78	2.47	-3.21	-0.938	-0.348	0.001	0.008
	D	Cyl.	0.466	2.25	3.95	-3.22	0.935	0.355	0.002	0.018
Pipe 3	M	Cyl.	0.46	-2.79	3.46	-3.10	-0.890	-0.445	-0.101	0.011
	D	Cyl.	0.39	2.87	6.13	-1.60	0.850	0.509	0.133	0.202
Pipe 4	M	Cyl.	0.46	-1.91	2.21	-2.56	-0.675	-0.458	-0.578	0.056
	D	Cyl.	0.446	0.63	3.88	-0.33	0.666	0.434	0.607	0.054

Table 5.9. Surfaces fit to real cylindrical data (a pipe of radius 0.46" in various orientations). Comparison of the performance of model-driven (M) versus data-driven (D) parameter extraction.

same. In all cases, the model-driven technique's best-fitting surface was a cylinder of radius 0.46". For the "Pipe 4" image, the data-driven technique resulted in a lower error than the model-driven technique, but the data-driven fitting recovered a slightly smaller radius parameter than the true value.

The model-driven conic fitting technique was applied to several scenes containing funnels whose vertex angle was approximately 0.53 radians. A profile plot of the range image of one of the images is shown in Figure 5.22. Another image of the same funnel, displayed using a perspective projection, is shown in Figure 5.23. The funnel in the latter image was occluded by a light-absorbing covering that prevented the range scanner from determining the depth of the occluded pixels. Table 5.10 summarizes the estimated parameters and classification of the conical patch segmented from Figure 5.22. The best fitting surface was a cone of vertex angle $\phi = 0.53$ radians. Table 5.11 shows the estimated parameters and classification of the conical surface patch segmented from Figure 5.23. Although the conic surface was partially occluded, the proper cone was accurately fit to the surface. The conical fit to the occluded funnel resulted in a larger error than the fit to the larger surface patch of Figure 5.10, however.

The classification method was also applied to several scenes of jumbles of parts and objects with multiple surface types. A pseudo-intensity rendering of a range image of an

Surf	ϕ	$\hat{v_x}$	$\hat{v_y}$	$\hat{v_z}$	$\hat{a_x}$	$\hat{a_y}$	$\hat{a_z}$	Error
Cone	0.53	-0.494	2.564	0.352	(-0.004,	-0.009,	-1.000)	0.004

Table 5.10. Classification and parameter estimation for funnel in Figure 5.22.

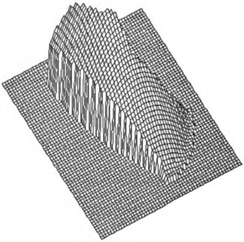


Figure 5.23. Depth profile of the range image of an occluded funnel.

Surf	ϕ	$\hat{v_x}$	$\hat{v_y}$	$\hat{v_z}$	$\hat{a_x}$	$\hat{a_y}$	$\hat{a_z}$	Error
Cone	0.53	0.046	3.191	0.307	(-0.001,	-0.006,	-1.000)	0.009

Table 5.11. Classification and parameter estimation for occluded funnel in Figure 5.23.

Figure

dering

pipe fit

object

using

regio

corre

hyp

par

dr

T

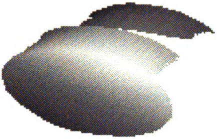


Figure 5.24. Pseudo-intensity rendering of range image of an adapter pipe fitting.

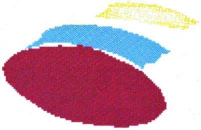


Figure 5.25. Segmentation of the adapter in Figure 5.24.

object composed of two cylinders is shown in Figure 5.24. A segmentation of that image, using the range image segmentation method of Flynn [55], is shown in Figure 5.25. Three regions were extracted, the planar base of the object and two cylindrical regions (with correct radii of 1.125" and 1.375").

The classification of the surfaces in Figure 5.25 is shown in Table 5.12. Only the hypothesized surface type and recovered radius parameter are displayed; the positional parameters are not shown [‡‡] The estimated parameters and classification by the model-driven technique is contrasted with the data-driven technique of Flynn and Jain [56]. The model-driven technique for cylindrical parameter estimation performed better for the smaller cylindrical surface. Both methods classified the other cylindrical patch quite well. The planar surface, as mentioned earlier, was fit using the data-driven technique [56].

Table 5.13 shows the performance of the fitting technique applied to the surface patches in Figure 5.27. The model database contained several spherical and conical surfaces (none of which fitted the surfaces in the image well) and cylindrical surfaces with radii of 1.0", 1.125", 1.25", 1.375", 1.75", 2", and 3". Note that there are several "voids" (small black regions inside surface patches) in the segmentation in Figure 5.27. The split-and-merge

[‡‡] It should be noted that the model-driven and data-driven techniques do not recover identical positional parameters.

Actua
Cy
(y
Cy
(

Table 5

of the a

Actual Surface	Parameter	Technique	Hyp. Surface	Est. Par.	Error
Cylinder (yellow)	r = 1.125"	Model-driven	Cylinder	r = 1.125"	0.006
		Data-driven	Cylinder	r = 0.948"	0.013
Cylinder (aqua)	r = 1.375"	Model-driven	Cylinder	r = 1.375"	0.014
		Data-driven	Cylinder	r = 1.31"	0.003
Plane	*	Data-driven	Plane	*	0.011

Table 5.12. Comparison of classification and parameter estimation of the surface patches of the adapter (Figure 5.25) by the data-driven technique and by our model-driven method.

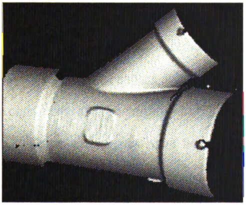


Figure 5.26. Pseudo-intensity rendering of range image of a Y pipe fitting.

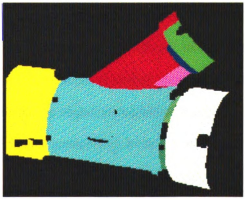


Figure 5.27. Surface segmentation of the Y pipe fitting shown in Figure 5.26.

A

Table
fitting

segn
then
seg
fitt
in
sl
w
r

Actual Surface	True Parameter	Hyp. Surface	Est. Par.	Error
Cylinder (red)	r = 1.25"	Cylinder	r = 1.25"	0.041
Cylinder (green)	r = 1.375"	Cylinder	r = 1.375"	0.019
Cylinder (yellow)	r = 2.0"	Cylinder	r = 2.0"	0.066
Cylinder (pink)	r = 1.25"	Cylinder	r = 1.125"	0.004
Cylinder (aqua)	r = 1.75"	Cylinder	r = 1.75"	0.024
Cylinder (white)	r = 2.0"	Cylinder	r = 2.0"	0.027

Table 5.13. Classification and parameter estimation of surface patches of the Y-shaped pipe fitting shown in Figure 5.27.

segmentation technique initially over-segmented the image, discarded tiny regions, and then merged regions. Most of the "voids" are small patches that were discarded by the segmentation scheme. Some of the voids were caused by missing range data, however. The fitting performed well for all the patches except the two small narrow patches (shown as indigo and gray colored patches). The pink patch was classified correctly but was fit with a slightly smaller cylinder (radius = 1.125"). The second-best hypothesis for the pink patch was a cylinder of radius 1.25", though. Hence, as mentioned earlier, in practice one could retain several high-ranking hypotheses for use by the object recognition module.

These techniques have been tested on images containing both convex and concave surfaces. One such image (see Figure 5.28) contained two concave spheres (the inside of two tennis balls) of radius 1.0" and one convex sphere (a rubber ball) of radius 1.21". All of the patches were classified correctly except for two patches that represent the fuzzy outer wrapping of the tennis balls. In the segmented image, Figure 5.29, these regions are shown in pink and yellow. The green, blue, and aqua concave surface patches in the segmented image were all classified as spheres of radius 1.0". Despite the segmentation of one of the concave surfaces into two patches, the estimated center positions for the spheres were

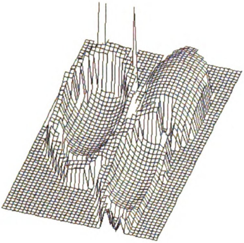


Figure 5.28. Depth profile of range image of two concave tennis balls and a convex rubber ball.

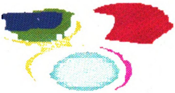


Figure 5.29. Segmentation of the balls shown in Figure 5.28.

within 0.035" of one another. The RMS errors in the fitted surfaces were less than 0.01 for the concave surface patches. The red patch, which is a (convex) rubber ball, was correctly classified as a sphere of radius 1.21" again with an RMS error measure of less than 0.01. For this image, the algorithm executed in approximately 27 seconds on a Sun SPARCstation 2 using a model database of 3 cones, 4 cylinders, and 3 spheres.

Actual Surface	True Parameter	Technique	Hyp. Surface	Est. Par.	Error
Sphere (red)	$r = 1.21$"	Model-driven	Sphere	$r = 1.21$	0.005
		Data-driven	Sphere	$r = 1.21$	0.005
Cylinder (green)	$r = 0.46$"	Model-driven	Cylinder	$r = 0.46$	0.004
		Data-driven	Plane	*	0.046
Cone (blue)	$\phi = .53$	Model-driven	Cone	$\phi = 0.53$	0.005
		Data-driven	Cylinder	$r = 1.65$	0.039
Cylinder (yellow)	$r = 0.39$"	Model-driven	Cylinder	$r = 0.39$	0.008
		Data-driven	Cylinder	$r = 0.34$	0.006

Table 5.14. Surface classification of the jumble of parts shown in Figure 5.31 showing comparison of model-driven versus data-driven parameter estimation. Only the top-ranked hypotheses for each surface is shown.

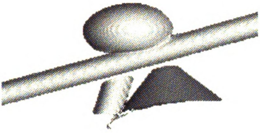


Figure 5.30. Pseudo-intensity rendering of range image of jumble of parts.

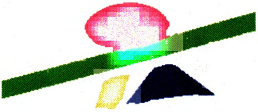
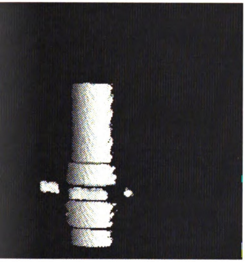


Figure 5.31. Segmentation of the jumble of parts shown in Figure 5.30.

A pseudo-intensity rendering of the range image of a part jumble is shown in Figure 5.30. The surface segmentation is shown in Figure 5.31. This segmentation contains four regions that correspond to a sphere of radius 1.21", a cylinder of radius 0.46", a cylinder of radius 0.39", and a cone of vertex angle 0.53 radians. In Table 5.14, the estimated parameters and classification of the patches is compared to Flynn's [56] data-driven technique. Both methods quite satisfactorily classified the sphere and the cylinder of radius 0.39". But the model-driven fitting provided much better classification for the cone and the second cylinder of radius 0.46". The reader should note that the data-driven fitting to which the method is compared cannot fit cones; it can only extract parameters for planar, spherical, and cylindrical surfaces.

Figure 5.32 contains a range image of a cylindrical casting. A depth profile plot of the cylindrical housing is shown in Figure 5.33. The surface segmentation of this image is shown in Figure 5.34. The segmentation technique employed included the extensions to the method of Flynn, Hoffman, and Jain discussed in Section 5.1. There are six regions in the segmentation, two of which (the blue and yellow regions) are non-quadric. The other

Fig
ran



gure 5.32. Pseudo-intensity rendering of nge image of cylindrical housing.

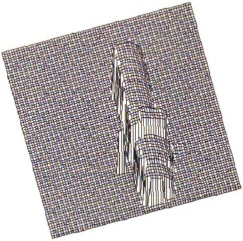


Figure 5.33. Depth profile plot of cylindrical housing image of Figure 5.32.

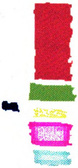


Figure 5.34. Segmentation of the image of the cylindrical housing in Figure 5.32 .

Actual Surface	True Parameter	Method	Hyp. Surface	Est. Par.	Error
Cylinder (red)	$r = 0.7875$”	M	Cylinder	$r = 0.7875$	0.005
		D	Cylinder	$r = 0.64$	0.020
Cylinder (green)	$r = 0.985$”	M	Cylinder	$r = 0.985$	0.008
		D	Cylinder	$r = 1.06$	0.019
Non-quadric (blue)	*	M	Plane	*	0.006
		D	Plane	*	0.006
Non-quadric (yellow)	*	M	Cylinder	$r = 0.985$	0.004
		D	Cylinder	$r = 10.29$	0.003
Cylinder (pink)	$r = 1.063$”	M	Cylinder	$r = 0.985$	0.004
		D	Cylinder	$r = 1.26$	0.010
Cylinder (aqua)	$r = 0.985$”	M	Cylinder	$r = 0.850$	0.006
		D	Cylinder	$r = 0.87$	0.011

Table 5.15. Classification and parameter estimation of surface patches of the cylindrical housing in Figure 5.34. Model- and data-driven fittings are contrasted.

four regions are all cylindrical. Table 5.15 shows the performance of the model-driven classification technique (indicated with an M) against the data-driven technique (indicated with a D). The model-driven fitting performed much better than the data-driven fitting. For the model-driven fitting, nine different cylinders were in the database. The radii of these cylinders were 0.50”, 0.7875”, 0.85”, 0.985”, 1.063”, 1.25”, 1.50”, 1.75”, and 2.00”. We attempted to fit spheres of several radii, also, but none of the spherical hypotheses fit the data very well. The parameters of two of the regions were recovered accurately by the model-driven fitting. For two other surfaces (the pink and aqua colored regions), the correct parametrization was the second best hypothesis. (The correct parametrization of the pink region had an RMS error measure of 0.0062. The aqua region’s correct parametrization had an RMS error of 0.0061.)

The acceptance of an incorrect parametrization of two of the cylinders in the image of the cylindrical casting illustrates a problem with the Technical Arts range sensor. The casting was oriented such that the sensor could only view (at most) one fourth of the cylinder. In the image, limb edges appear at the top of the cylinder and at the side of the cylinder. The shaded area in Figure 5.35 shows the portion of the cylinder that is visible to the sensor in

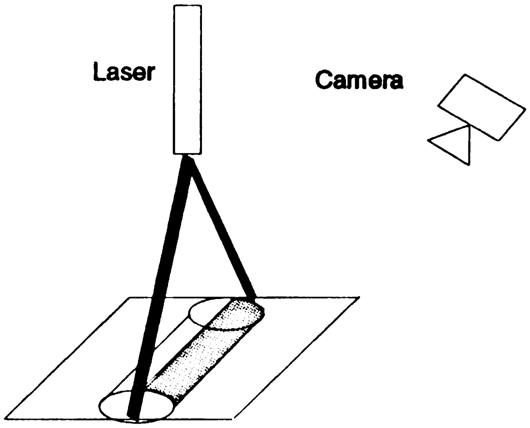
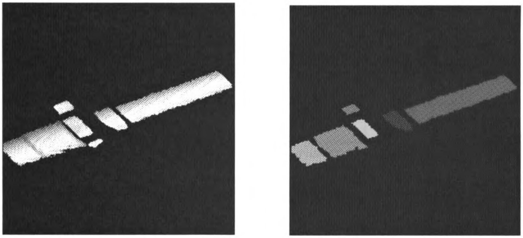


Figure 5.35. Diagram of the sensing arrangement for Figure 5.32.

the image of Figure 5.32. The depth values returned by our range sensor near the cylinder limb whose normals are oriented in the horizontal (x-y) plane are very noisy. This effect results because the sensor estimates depth near edges by averaging several points whose depths vary significantly. The estimation of the position of where the beam is brightest introduces the error for those values. As a result, our cylinder-fitting technique cannot recover the position of the cylindrical axis as accurately for this orientation. The next example, in Figure 5.36, illustrates that if the cylinder is rotated 90° about the z axis, then the parameters can be recovered accurately. This is because the depth values near the two limbs "cancel out" one another.

The casting in Figure 5.36 is oriented such that the range scanner can sense a larger surface area of the casting. Instead of observing a limb edge at the "top" and "side" of the cylinder, now both the limb edges are observed at the "side" of the cylinder. Unfortunately, these limb edges are not the ideal limb edges one would observe in a backlit intensity image. Furthermore, the edges are very noisy. Thus, jump edges found in range images from the White Scanner are problematic and probably ill-suited for classification or recognition purposes. For classification to be acceptable, the object must be oriented such that the range sensor can image a significant amount of the surface. Therefore, we restrict the position



(a) Range image.

(b) Segmented image.

Figure 5.36. Cylindrical housing image and its segmentation.

of the object to be such that the sensor can acquire an acceptable image. Alternatively, we could collect images from several viewpoints and merge the data or invest in a better sensor.

Table 5.16 indicates the model-driven classification for the surfaces in Figure 5.36(b). The same model database was used as in the previous example (the image of Figure 5.34). A data-driven fitting performed worse for every surface, either by fitting incorrect parameters, or by fitting nearly correct parameters but with much larger errors than the model-driven fitting. Our fitting is, in general, better than the fitting for the object in Figure 5.34. Region 3 (the blue region), is very small and as a result a cylinder of radius 0.85" fits better than the correct cylinder of radius 0.985" (the 0.985" radius cylinder is the second-best hypothesis and has RMS error 0.0091).

There is still some sensor noise near the limb jump edges, but the noise from the two limbs "cancels out" because the Hough technique that accumulates evidence for the axis position is able to average the off-center votes from each side and thus generate reasonable axis position estimates. Larger local deviations from the fit surface are observed near the limb edges than near the center of the regions, as is illustrated in Figure 5.37.

The diagram in Figure 5.37 shows the actual surface of the casting, the fitted surface,

Actual S
Cylir # 1 (
Non-q # 2 (
Cyl # 3
Non # 4
C #
(#

Table 5.16.
fitting techr

Actual Surface	Parameter	Hyp. Surface	Est. Par.	Error
Cylinder # 1 (red)	$r = 0.7875"$	Cylinder	$r = 0.7875"$	0.0053
Non-quadric # 2 (green)	*	Plane	*	0.0039
Cylinder # 3 (blue)	$r = 0.985"$	Cylinder	$r = 0.85"$	0.0070
Non-quadric # 4 (yellow)	*	Cylinder	$r = 0.7875"$	0.0079
Cylinder # 5 (pink)	$r = 1.063"$	Cylinder	$r = 1.063"$	0.0042
Cylinder # 6 (aqua)	$r = 0.985"$	Cylinder	$r = 0.985"$	0.0052

Table 5.16. Classification of surfaces of cylindrical housing in Figure 5.36 by model-driven fitting technique. Only the top-ranked hypothesis for each surface is shown.

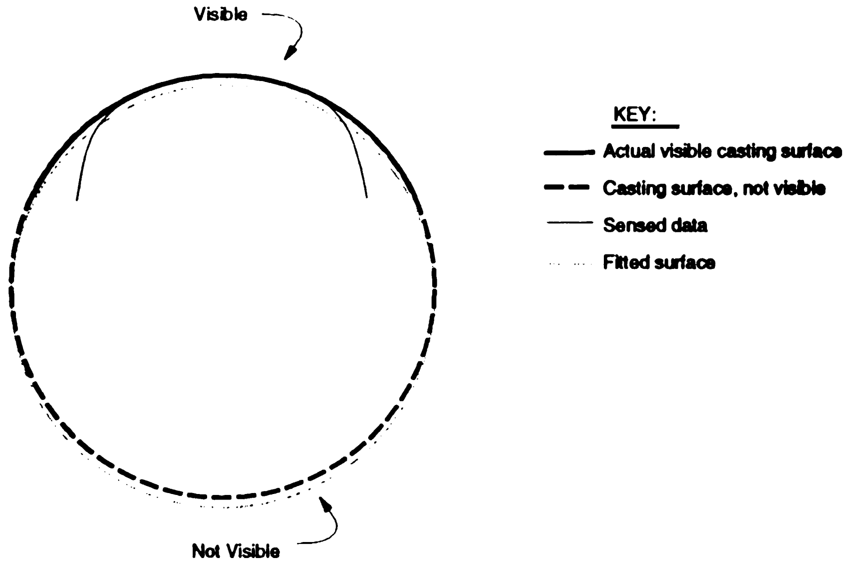


Figure 5.37. Cross-sectional view of cylinder showing deviation of sensed data from the fitted cylindrical surface. The diagram is exaggerated to illustrate the sensor problem.

and the sensed data. We observe that the sensor regularly delivered cylindrical data in this form. The deviation in fit in many images followed this same pattern of very large negative deviations near the limb edges and small positive deviations near the "top" of the cylinder.

5.3 Conclusions and Extensions

Segmentation of the castings is accomplished by using a revision of Flynn, Hoffman, and Jain's local feature clustering and merging algorithm. Further extensions to the method are being explored to allow more model information to be incorporated into the segmentation scheme. Ideally, an initial coarse segmentation could be used to provide rough evidence for the presence of certain model surfaces. The model could then be used to directly impose segmentation on the data.

The suite of classification techniques extracts quadric surface parameters by incorporating geometric characteristics of the surface type and invariant parameters directly into the fitting procedure. The methods are model-driven and use Hough Transform techniques for cylindrical and spherical fitting. Many experiments on real and synthetic data containing concave and convex surfaces demonstrate the utility and accuracy of the techniques.

The size of many of the quadric surfaces on castings is small, making both segmentation and classification difficult problems. We also experienced difficulty segmenting and classifying filleted surfaces on the castings. These problems could be alleviated by making more use of model information during segmentation. It is also possible to use the output of the current segmentation and classification as an input to a second segmentation. The re-segmentation could use a coarse estimate of pose arising from matching the reliably labeled surfaces of the first segmentation and classification to the model. The coarse pose estimate could be applied to the model and the model's labelings could then be imposed onto unlabeled regions in the range image. This new segmentation could be used to compute a better estimate of pose. It might be necessary to re-segment some regions using the model-imposed labels, however. We are currently working on model-based segmentation.

One weakness of the conic fitting procedure is its two-step sequential nature. Errors that arise in the first step of vertex location determination are propagated into the second step (axis vector estimation). There are probably enhancements that can be made to the algorithm to reduce the effects of this error. It was difficult to fit cones with vertex angles less than 15 degrees perhaps due in part to the two-step process.

The technique could execute faster by using a random sample of the data points for hypothesis generation and/or for verification, similar to the Randomized Hough Transform [189] and the RANSAC method [53]. Parallel computing could also be used to speed up the performance. The Hough voting scheme could be implemented in parallel, for instance, on a machine such as the Connection Machine. Using a subset of the data and using parallel computing should result in very fast extraction of surface parameters. On a coarser level of parallelism, several different surfaces could be fit at the same time by different teams of processing elements where each team of elements is devoted to extracting parameters for one type of surface in parallel.

Currently, only planes, spheres, cylinders, and cones can be fit, so extensions need to be made to allow fitting of other surfaces. It would probably be advantageous, though, to develop an algorithm for filleted surface fitting, provided that filleted regions can be segmented reliably and that they are large enough to allow reliable fitting. Ideally, a free-form surface fitting algorithm could be developed to fit all surfaces. But free-form surface fitting is a difficult problem that has only recently begun to receive substantial attention. The algorithms we have developed execute quickly and seem ideal candidates for implementation on SIMD or systolic architectures. Parallel implementations could be used to achieve real-time execution.

CHAPTER 6

Object Localization

In the inspection algorithms of this research project, we generally assume that the object is in its correct pose (i.e., both position and orientation), thus bypassing the object localization problem. In the context of the tree diagram of Figure 1.8, our approach has thus followed the leftmost branch of each level of the tree down to the leftmost terminal node. In fact, constraining object pose is a reasonable assumption on many assembly lines. Operating with constrained pose is analogous to inspecting in a well-jigged (well-fixtured) environment. However, special-purpose fixtures are expensive and probably only cost-effective when large volumes of parts are being produced. Ideally, we would like to relax the pose constraint to increase the flexibility of our automated inspection algorithms. In the inspection algorithms presented in this dissertation, we assume that the correspondence between model and object surfaces has been established, perhaps using one of the techniques presented in this chapter, and that it is necessary to inspect the object surfaces. An explicit estimate of pose is not necessary for the surface shape and feature inspections of Chapters 8 and 9, although a pose estimate is necessary for the gross feature inspection of Chapter 7. One difficulty in using estimated transformations is that these typically introduce positional errors that complicate inspection. In this chapter, several techniques for determining pose are presented.

6.1 M

One appro

presented

vary grea

used, an

of a par

this is

A pho

by a n

objec

A

visio

text

pre

de

m

p

C

6.1 Mechanical solutions

One approach to the object localization problem is to assume that the object is always presented to the sensor in approximately the same position (although the orientation may vary greatly). This is a reasonable assumption because simple mechanical probes can be used, and have been used for some time in industry, to determine the presence or absence of a part within the visible envelope of the imaging sensor [99]. (A simple illustration of this is a beam of light projected 1 inch above an assembly line from one side to another. A photovoltaic sensor opposite the beam projector detects when the light beam is broken by a moving part. When the beam is broken, the range sensor can acquire an image of the object.)

A number of techniques have been proposed for 3D object localization in the computer vision literature. We have investigated several techniques for pose estimation in the context of object inspection. Our current system assumes that the objects for inspection are presented in constrained poses through fixturing or that a reliable localization method has determined their approximate pose. Because of the limitations of our sensor, precise estimation of pose may only be possible through fixturing. Only the inspection techniques we present in Chapter 7 require precise pose estimation, however. The inspection techniques of Chapters 8 and 9 merely require that the correspondence between object and model surfaces has been established.

6.2 Overview of Approaches to Localization

Several papers on part position localization have been presented in the literature. Gunnarsson and Prinz [66], for example, have presented a scheme for locating polyhedral objects. They claim that their method can also be extended to arbitrary shapes. Their method is an iterative technique to find an optimal transformation matrix. Stockman [162] has proposed pos

a generalized

detecting a

recognition

using a sea

recognitio

Techniqu

is discuss

of attent

An

viewpo

relatio

of the

grap

prop

Fre

Fre

6

T

-

proposed pose clustering for localization of objects. His technique, which is similar to a generalized Hough transformation, determines a globally acceptable transformation by detecting a cluster in the space of all coordinate transformations. Most of the 3D object recognition approaches compute object location simultaneously with recognition* usually using a search technique. A short discussion of some of these approaches to 3D object recognition is presented later in this chapter in the discussion of interpretation tree search. Techniques based on the Extended Gaussian Image (EGI) [78] have also been pursued. EGI is discussed in Section 6.3.2. Aspect matching techniques have also received a great deal of attention.

An *aspect* is a viewer-centered representation that models object appearance from one viewpoint [174]. The aspect graph consists of a collection of views, or aspects, and the relationship between the views. Usually, the nodes in the graph represent ***general views*** of the object. A general view is a set of viewpoints that are equivalent. The links in the graph represent the transitions between adjacent views. The aspect graph concept was first proposed by Koenderink and Van Doorn [100] who termed it *visual potential.* Chen and Freeman [30] and Bowyer et al. [19] have also investigated aspects, although Chen and Freeman prefer the term *characteristic views.*

6.3 Techniques for Pre-Segmentation Localization

This section presents techniques for pose determination without (or before) segmentation. Two of the techniques for this include the EGI method and matching to stored views (aspects). Several methods for pre-segmentation localization were investigated in our research. These methods all appear to have some limitations. The most reliable of these methods appears useful only for resolving 2D rotational transformations, or very small 3D rotational transformations (e.g., less than 3-5 degrees).

*Our focus in this dissertation is on inspection, not recognition, however.

6.3.1 Mat

We attempte

object to stor

This technic

computatio

difficult ta

several po

This

The corr

anywhe

ranking

metho

a poo

appr

Bo

Tw

in

a

p

6.3.1 Matching to Stored Views

We attempted to determine a coarse estimate of pose by matching a sensed image of an object to stored views of the object model using the matching scores presented in Chapter 7. This technique is similar to aspect graph matching, although our approach avoided explicit computation of the general views. Computing the general views of the aspect graph is a difficult task. We matched real and synthetic range images against 500 stored views of several polyhedral objects and one object with curved surfaces.

This technique was not able to determine a good initial estimate of pose, however. The correct view usually did not exhibit a high matching score. The correct view ranged anywhere from the highest matching score to the fortieth largest score, with a typical ranking of tenth to fifteenth largest. Thus, we abandoned this approach and examined other methods for pose determination. It should be noted that stored view matching may only be a poor approach for polyhedral objects (since we did not perform exhaustive testing of the approach for non-polyhedral objects); view matching may be acceptable for other objects.

Boundary Alignment - Chamfering and Hausdorff Matching

Two techniques for matching the silhouette of an object to the silhouette of a model were also investigated. These techniques require the viewing direction to be known; the techniques are restricted to 2D rotations![†] Our experience with these techniques has been that they produce good results (rotation can usually be recovered with 0.5° accuracy) within this constraint. However, we were unable to reliably detect the approximate viewing direction (through EGI or stored-view matching), so these techniques are not currently useful for our inspection system.

The first technique is the chamfering algorithm implemented by Chen and Stock-

[†] The chamfering method has been extended to have a limited capability to recover 3D rotational parameters [28]. This extension requires the use of multiple stored views. It also requires that the approximate pose be known (that is, the actual pose must be a small rotation from one of the stored views). We have only used the chamfering method to recover 2D rotations, however.

man [28]. The chamfering technique measures the difference between the object and the model silhouettes. The chamfering expands each point in the edge map in all directions. For each pixel in the expansion, the distance of that pixel from the edge is stored. The set of distances forms a distance function. Silhouettes can be aligned by minimizing the distance function.

The second technique is the Hausdorff matching [82]. We used Pankanti et al.'s [135] implementation of the Hausdorff matching to recover x and y translations and rotations about z. The Hausdorff matching is used in [135] to compare the "distance" between model and scene edge maps. A constrained search strategy is used to find the best transformation that aligns the model with the scene. The boundary of the model can be viewed as a point set P that must be aligned with the transformed point set Q that represents the boundary of the object in the scene. The Hausdorff matching between P and Q measures the proximity of the edge points when no correspondence between points in P and Q is known. Formally, the Hausdorff distance is the max-min distance

$$H(P,Q) = \max(h(P,Q), h(Q,P)),$$

where

$$h(P,Q) = \max_{p \in P} \min_{q \in Q} \| p - q \|,$$

and $\| \cdot \|$ is the Euclidean norm. To allow for some occlusions, the Hausdorff matching scheme did not require a matching of all of the points. A typical threshold for matching would use 85 to 95 percent of the points.

Both the chamfering and Hausdorff matching were able to resolve 2D rotations within 0.5°. Our application of Hausdorff matching for inspection is described in Chapter 7.

6.3.2 EGI

There are a number of descriptions in the literature of the Extended Gaussian Image (EGI), for pose determination (and object recognition) [33, 70, 78, 112]. The EGI was first proposed by Horn and is described in detail in his book [78]. The EGI of an object is a unit, or Gaussian, sphere whose surface contains the distribution of the object surface-orientation normals. The Extended Gaussian Image represents an object as the collection of its unit-length surface normals. The surface normal at each point on the object is translated to the origin and assigned a unit weight. The Gaussian sphere is then *extended* by storing, at each position on the sphere that corresponds to the head of a surface-orientation normal vector, a value equal to the number of surface normals oriented in that direction. For example, a planar surface with unit-length surface orientation normal vector $\mathbf{n} = (n_x, n_y, n_z)$ would cause a weight proportional to the plane's surface area to be stored at (n_x, n_y, n_z) on the Gaussian sphere. The EGI is actually a surface normal vector histogram in three dimensions.

The EGI is also related to the collection of Gaussian curvatures K of the object. For instance, if δO_i is a small surface region on the object and if δS_i is the corresponding region on the Gaussian sphere, then Gaussian curvature

$$K = \lim_{\delta O_i \to 0} \frac{\delta S_i}{\delta O_i} = \frac{dS_i}{dO_i},$$

where O_i is a finite region of the object and S_i is the area of the corresponding region of the Gaussian sphere. The EGI is $G(\xi, \eta) = \frac{1}{K(u,v)}$, where (ξ, η) is the Gaussian sphere point that corresponds to object surface point (u, v).

An object can be modeled by constructing the extended Gaussian sphere, usually supplemented by additional features about each possible viewing direction on a uniformly sampled viewing sphere. The additional features typically encode the total amount of surface area visible from the viewpoint.

Object recognition (or pose determination) consists of matching the EGI of the sensed object with the EGI of the stored model(s). The object and model EGI centers are made coincident and the model EGI is then rotated about (each) candidate line of sight until a position of minimum difference between the mass distributions in the EGIs is reached. Candidate viewpoints are usually selected using a heuristic based on the additional features stored about each of the possible viewpoints.

The EGI is usually modeled as an approximation of a tesselated sphere. The masses of the surface orientation normals are accumulated in the tesselated sphere's cells and stored along with the supplemental matching features. Unfortunately, there are no known methods to create a uniform tesselation on a sphere. Instead, the EGI methods use polyhedral approximations of tesselated spheres.

Horn [78] has identified the ideal characteristics of a tesselation to be used for an EGI. Ideally, any tesselation should have cells that have the same area and shape and which occur in a regular pattern. Also, there should be enough cells to allow for fine resolution in pose determination. Finally, the cells should have somewhat rounded shapes. Unfortunately, there is no known tesselation that meets all of these goals. Only the five platonic (or regular) solids (tetrahedra, hexahedra, octahedra, dodecahedra, and icosahedra) have regular polygons as faces with each of the polygons identical on the solid. All of these tesselations lack the resolution necessary for object recognition and/or pose determination.

Instead, polyhedra which are semiregular are commonly adopted for use as an approximate tesselated sphere. Most of the applications of EGI have used polyhedra constructed from dodecahedra or icosahedra. The faces of these platonic solids are subdivided into equal-sized triangles and then projected onto the sphere to produce a geodesic dome that closely approximates a uniform tesselation of the sphere. The "cells" of the discrete EGI are the faces of these projected polyhedra. A 320-cell tesselated sphere created by subdividing an icosahedron is shown in Figure 6.1. In practice, the dual of the geodesic dome is preferred to the geodesic dome to create cells that are more "rounded" in shape. The dual can be constructed by defining a new polyhedron formed by connecting the centers of

F

ac

al. (

the

to de

V

sever

match

were .

of Cha

polyhe

We

nation o

the best

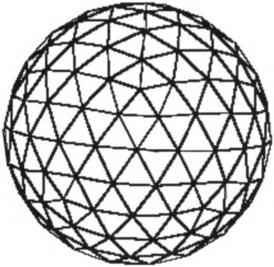


Figure 6.1. Tesselated sphere (frequency 4 geodesic polyhedron derived from icosahedron).

adjacent cells in the geodesic dome.

Horn [78], Caponetti et al. [24], Chen and Kak [26], Nagata and Zha [123], Shirai et al. [155], and Yang and Kak [190] have all developed object localization systems that utilize the Extended Gaussian Image. Nayar et al. [124] have used features extracted from the EGI to detect defects in solder joints.

We have implemented the EGI technique for pose determination. We built EGIs of several models of polyhedral blocks, one of which is shown in Figure 6.2 and tried to match synthetic and real range images to the EGI model. (The synthetic range images were synthesized from a CAD model of the block using the program discussed at the end of Chapter 4.) Our orientation histogram contained 320 cells; the frequency 4 geodesic polyhedron derived from an icosahedron was used.

We initially attempted to use a heuristic proposed by Yang and Kak [190] for determination of viewing direction and then rotated the model about the viewing direction to find the best match between model and object. It is important to use a good heuristic to limit the

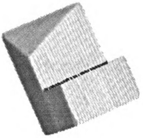


Figure 6.2. Pseudo-intensity rendering of polyhedral block.

number of candidate viewing directions. Otherwise, the search will take a very long time. The heuristic that Yang and Kak used was based on the ratio of visible surface areas to the projected area onto the image plane. This type of constraint is common for the EGI-based efforts reported in the literature. Yang and Kak computed a statistic R_{pa} for both the object and model EGI to compute the most likely viewing direction. For the model EGI, an R_{pa} was computed for each cell of the sphere as follows:

$$R_{pa} = \frac{\sum_{i=1}^{m} \vec{M}_i \cdot \vec{V}}{\sum_{i=1}^{m} |\vec{M}_i|},$$

where m is the number of cells in the EGI, $\vec{V}$ is the viewing direction, and $\vec{M}_i$ is the ith cell in the EGI with direction and magnitude determined by the model. The dot product $\vec{M}_i \cdot \vec{V}$ gives the first moment of mass of the visible hemisphere of the model. For the object EGI, a single R_{pa} statistic was computed:

$$R_{pa} = \frac{n}{\sum_{i=1}^{n} [\frac{1}{\cos\theta_i}]},$$

where n is the number of visible patches and θ_i is the angle between the surface normal of the ith patch and the viewing direction (which is the z axis). We found that a bound of 0.1 was necessary in matching R_{pa} scores and that this only eliminated half of the potential viewing directions.

v

w

m

we

0.1,

sma

I

whic

matcl

mode

W

Specifi

the surf

The

The matching score between model and object EGI was computed using

$$\sum_{i=1}^{n} \| c_i - M_{i+d} \|^2,$$

where c_i denotes the contents of object cell i and M_{i+d} denotes the contents of the cell $i + d$ in the model that is closest to object cell i (i.e., d is a minimum). Cells that were not visible from the candidate viewing direction were not considered. The correspondences that produced the forty smallest matching scores were retained.

Eventually, we abandoned the heuristic of R_{pa} because the correct pose hypothesis was usually not among the twenty highest-ranked matches. We did not choose the viewing direction explicitly after abandoning R_{pa}. Instead, we matched the tessel in the object EGI that had the highest weight with every non-empty cell in the model, matching the highest weight model EGI cell first and then proceeding to the second largest, down to the smallest weight model cell. For each of the matchings between the object and model tessel, the model was rotated about its matched tessel. This generated a series of hypotheses which were ranked according to their matching scores.

Furthermore, we removed all object EGI cells that had a small z component (less than 0.1) because they were highly likely to result from sensor noise. Object cells with a very small weight (i.e., corresponding to a surface area of less than 0.01) were also discarded.

In the computation of the matching score, the best matching model cell was taken as whichever cell within 10° of the central angle of the object EGI cell that most closely matched the object cell. (This is in contrast to the original technique of only choosing the model cell whose distance is closest.)

We also "blurred" the model EGI by overlapping the cells during model EGI generation. Specifically, every cell within 10° of an ideal normal $\vec{n}$ was incremented with the weight of the surface associated with normal direction $\vec{n}$.

The EGI matching procedure executed slowly and did not deliver very good results *i*

c

2

hy

fo

EC

EG

mo

listi

the

corr

hypo

W

by *Ha*

it does

EGI is

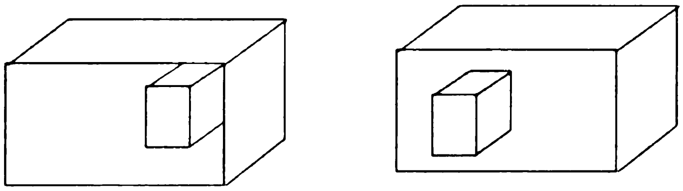


Figure 6.3. Two objects that have the same EGI representations.

even after these enhancements. Execution of the matching took between 5 and 10 minutes of CPU time on a SPARCstation 2. Generation of an EGI took less than 30 seconds. The correct hypothesis usually was fairly low in the ranking of hypotheses. For five range images of the polyhedral block of Figure 6.2, the correct hypothesis was the third highest ranking in one case, fifth highest in another, sixteenth in another, and twentieth in two others. Pose was estimated within 3° in one case, 5° in another, and ranged from 10° to 20° in the others. For another polyhedral block, the matching was even poorer. The correct hypothesis was the tenth highest ranking in one of the two instances tried and was the fortieth in the other.

We experimented with "blurring" the object EGI in the same manner as the model EGI was blurred, but found that, in general, it did not improve results. We tried object EGI blurring both with and without model EGI blurring. Object EGI blurring without model EGI blurring produced little changes in the position of the correct hypothesis on the listing of best-ranked hypotheses. When coupled with model EGI blurring, the effect on the position of the correct hypothesis was also generally neutral, although in one case the correct hypothesis dropped from twentieth to below fortieth and in another case the correct hypothesis improved from sixteenth to first.

We are pessimistic about the EGI method, however, sharing the reservations expressed by Haralick and Shapiro [70]: the Gaussian image is not invertible for planar objects and it does not represent objects precisely enough for use in most vision tasks [70]. Also, the EGI is only unique for convex solids [112]. An example of two objects that have identical

EGI representations is shown in Figure 6.3.

Several additional problems with the EGI method have been presented in the literature [78] and have been confirmed by our experience. First, when the model EGI is rotated about a viewing angle, few of its rotated tessels exactly match tessels of the object EGI. This is caused by the non-uniform size and irregular distribution pattern of the cells. Also, it is very laborious to resolve the rotation about the viewing direction due to the large number of potential correspondences. It might appear that the EGI method could be implemented in a hierarchical manner, but most hierarchical approaches have not worked well because an object's appearance can change dramatically over only a small change in viewing direction. Our experiments verified this. EGI's greatest limitation for inspection purposes may be that it can, at best, determine pose within 5° to 10°, which is far too coarse for practical application. Finally, the general application of EGI is limited because there is no known way to generate the EGI for objects that are more "complicated" than objects that only have planar, spherical, and cylindrical surfaces.

6.4 Techniques for Post-Segmentation Localization

Probably the most successful techniques for object localization require that a reasonable segmentation of an image be available. Most of the techniques for post-segmentation localization have at least some capacity to tolerate small segmentation errors, although occasionally the trade-off is a decreased ability to determine the pose accurately.

6.4.1 Interpretation Tree Search

Most object recognition systems in the literature have determined object position and orientation either during, or as a result of, the recognition process. The transformation from model to the sensed view can be determined using the *interpretation tree (IT) search* [64, 65] method for matching model surfaces to sensed object surfaces. The interpretation tree

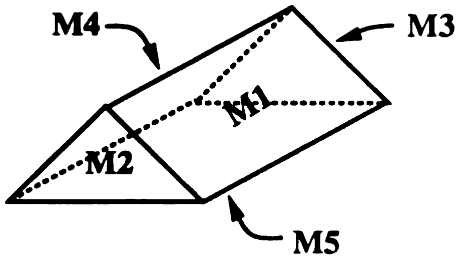


Figure 6.4. Wireframe model of a prism with surfaces numbered.

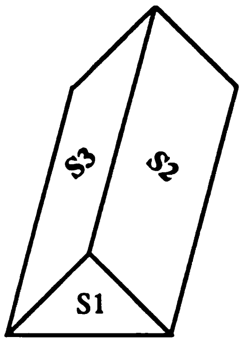


Figure 6.5. Visible surfaces on the prism with surfaces numbered.

paradigm enumerates all of the possible mappings between features on a model with features derived from a sensed object. The interpretation tree can be searched to find one or more consistent mappings from a model to the sensed object. A variety of constraints and heuristics are commonly used to prune the tree in order to reduce the computational complexity of the search.

Each node in the interpretation tree represents an association between one scene feature and one model feature. Object recognition can be viewed as matching an object's features against an interpretation forest consisting of a tree for each object model in the database. The set of valid mappings from the object features to the corresponding model features that survive the pruning constraints form a collection of hypotheses indicating which object is present in the scene. Each valid mapping also contains evidence for the position of object features vis-à-vis model features. An interpretation tree search is, therefore, useful for recognizing objects and for determining the rotation and translational parameters which transform the hypothesized model onto the sensed object. In practice, the interpretation tree usually has to be at least four levels deep to allow accurate estimation of the rotational and translation parameters.

A diagram of an interpretation tree is provided in Figure 6.6. In this tree, the features being matched are surface regions. The sensed object, shown in Figure 6.5, has three visible

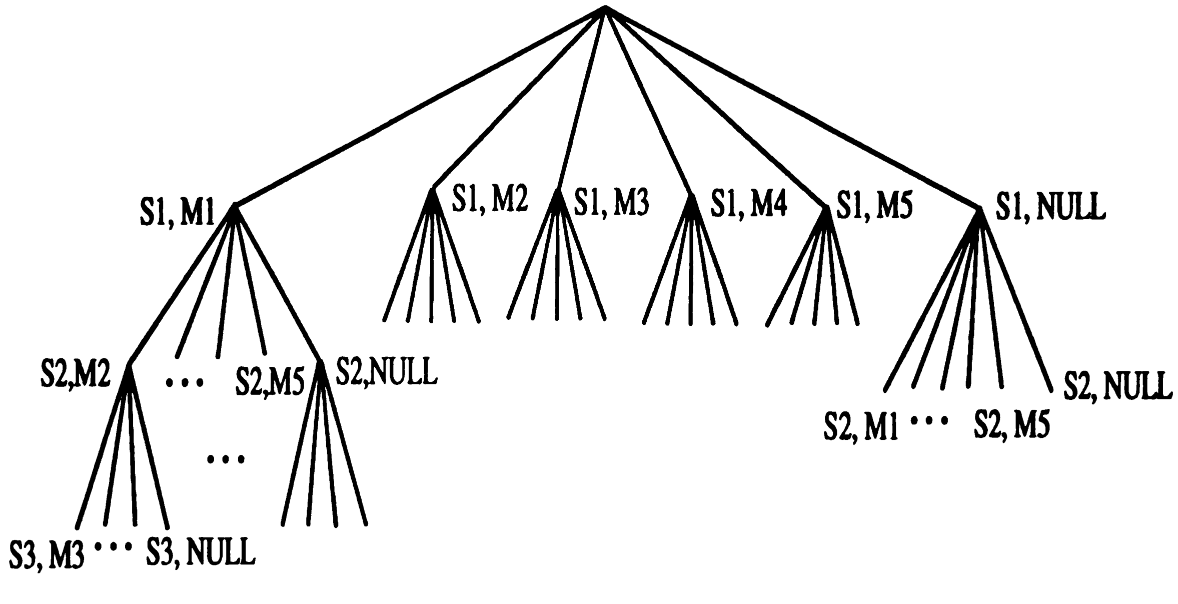


Figure 6.6. Interpretation Tree for an image with 3 visible surfaces matched to a model with 5 surfaces.

surfaces and the model, shown in Figure 6.4, has five surfaces. A surface region is defined as a patch of the object which is homogeneous in some sense. For instance, on the prism in Figure 6.4, there are five regions and the surface normals are all identical within a region. In Figure 6.6, object features can also be matched to a model feature labeled as a "wildcard" or "NULL" node, which is useful for handling any spurious features in a sensed image. Spurious regions could result, for example, from sensor noise which caused the violation of the homogeneity criteria. This might cause additional (false) regions to appear in an image which have no analog in the model. Allowing these types of regions to map to a "NULL" feature is an important aspect of the matching procedure because both sensors and low-level vision algorithms generally produce imperfect results. Unfortunately, the wildcard match feature causes many searches in practice to approach exponential complexity [64, 54]. Fisher [54] has recently developed an alternate method of generating the search tree that reduces the expected size of the search tree, although its worst case complexity is still exponential. In Figure 6.6, the top level of the tree matches model surfaces (denoted M1, *I*

a

E

to

we

mo

Pa

Bec

time

tasks

imag

Conn

in the

or ver

who h

M2, M3, M4, M5, and NULL) to image surface S1. The second level denotes the matching of model surfaces to image surface S2. The third level, most of which has been omitted from the diagram, denotes the matching of model surfaces to image surface S3.

There is substantial parallelism inherent in the interpretation tree search and even more parallelism in a search of a forest of interpretation trees (one tree per object model). The problem of object recognition can be posed as a search of a forest of interpretation trees. Each of the nodes in the tree or forest can be mapped onto a processing element that evaluates the pruning constraints for that node. If there are multiple constraints to be evaluated, each constraint for the node could be evaluated by different processing elements. The nodes need to be evaluated in a top-down manner, since if any constraints are violated for a node, then it is unnecessary to examine any descendants of the node in the tree.

Interpretation tree search has been used as the matching strategy for object recognition and subsequent pose determination in many vision systems ([58],[59],[64],[83], and [111]). Even if the identity of the object is already known, interpretation tree search can be used to determine the mapping from model surfaces to object surfaces [125]. In this research, we have used interpretation tree search to generate these mappings and to determine the model-to-view transformation (i.e., to estimate pose).

Parallel IT search

Because inspection requires a large amount of data to be processed in a short amount of time, serial processors may be inadequate for real-time processing. Most computer vision tasks have been pursued on sequential machines. Some low-level vision tasks, such as image convolution, are ideally suited for implementation on SIMD machines, such as the Connection Machine-2, or on systolic architectures [36] and have received some attention in the literature. Parallel approaches to higher level computer vision tasks such as matching or verification have received far less attention. An exception is Bhanu and Nuttall [13], who have implemented a 3D object recognition system on an 18-node BBN Butterfly.

Bhanu and Nuttall's technique is an MIMD implementation of a clustering algorithm for classification of quadric surfaces, such as spheres, cones, and cylinders, using curvature information from range data. Bhanu and Nuttall's work used mostly synthetic range data. However, their results are significant because the parallel implementation achieved near-linear speedup. They observed a maximum relative speedup of about 8 when using 12 processing elements. The speedups depended on the specific object, however. Few details of their implementation are known to us.

Our Approach to IT Search

In the remainder of this section, we describe our work on model-to-view transformation determination (and object recognition) using IT search for objects constructed of planar or quadric (spherical, cylindrical, or conical) surfaces. A parallel technique is then discussed which we have used for object recognition and pose determination. In this section, we focus on IT search for polyhedral objects, although the serial version of our IT search can also handle objects with quadric surfaces. We wish to demonstrate that the IT search is appropriate for parallel implementation. Parallel computation may be necessary for real-time automated visual inspection.

The features matched in our interpretation tree are the surfaces of the 3D objects. We used characteristics of each planar face as constraints to prune the search tree[‡]; a match between image surface i and model surface j was only considered if the constraints were satisfied. The facet constraints which were used for polyhedral objects included surface area, surface adjacencies, vertex angles of planar surface vertices, visibility attributes, and angles of intersection between surfaces. Surface area and the edge intersection angles can be considered to be unary constraints. The other characteristics were binary constraints that

[‡] As mentioned earlier, we have implemented a version of interpretation tree search on the Sun SPARCstations that can handle objects with quadric surfaces. That version uses the curved surface pruning constraints in addition to the constraints for planar surfaces. Specifically, the visible surface area, surface type, and surface parameters for spherical, cylindrical, and conical surfaces are used.

captured inter-surface relationships. Finally, we eliminate any surviving hypotheses that do not have at least 3 surface correspondences.

A complete matching between all image surfaces and corresponding model (or NULL) surfaces was considered an hypothesis which must be verified. The verification step computed the total angular difference between the mean surface normals of sensed object surfaces with the estimated surface normals of the transformed model. We then accepted the hypothesis which had the smallest total angular difference. A better verification approach would be to use the matching statistics proposed in Chapter 7.

The correspondences between image and model features are used to estimate the rotation transformation necessary to orient the model with the object. We then estimate the translation necessary to make the position of the rotated model coincident with the position of the object.

We first report our results of using IT search on a serial machine and then discuss the feasibility of implementing the search on a parallel machine.

A set of four images of a defective object is presented to demonstrate the localization capabilities of IT search in the presence of defects. The images of the defective object and their segmentations are shown in Figures 6.7, 6.8, 6.9, and 6.10. The defects in this case were created by stretching rubber bands around a polyhedral object we call Block2. In general, our technique for pose estimation performed quite well. Visual results indicate that we can generally determine pose within one or two pixels.

For the object in Figure 6.7, we were unable to find the correct pose estimate because the two small patches on one side of the block were not matched to any of the model surfaces. Our interpretation tree search uses edge features as a pruning constraint and this constraint was violated for the patches resulting from the segmentation. Edge features may be unreliable in the presence of defects. This is unfortunate, because edge features are a strong pruning constraint in our interpretation tree search. For instance, for one of the defect-free Block2 objects, the use of edge features reduced the number of hypotheses

(a) Range image of a defective Block2 object.

(b) Segmentation of the image in (a).

Figure 6.7. A range image of a defective instance of Block2. A rubber band has been tied around the block to generate the defect.

(a) Range image of a defective Block2 object.

(b) Segmentation of the image in (a).

Figure 6.8. A range image of a defective instance of Block2. A rubber band has been tied around the block to generate the defect.

(a) Range image of a defective Block2 object.

(b) Segmentation of the image in (a).

Figure 6.9. A range image of a defective instance of Block2. A rubber band has been tied around the block to generate the defect.

(a) One view of a defective Block2 object.

(b) Segmentation of the image in (a).

Figure 6.10. A range image of a defective Block2. A rubber band has been tied around the block.

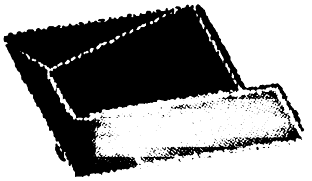


Figure 6.11. Wireframe of transformed model overlaid on sensed range image of Figure 6.8.

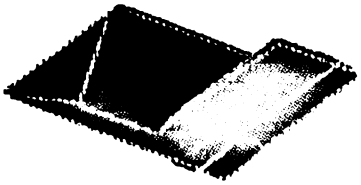


Figure 6.12. Wireframe of transformed model overlaid on sensed range image of Figure 6.9.

which survive pruning from 280 to 151.

However, for Figures 6.8 and 6.9, our pose estimate was quite good. We have overlaid a wireframe of the model on the sensed images in Figures 6.11 and 6.12. In each case, the model was transformed using the rotation and translation estimates generated by the interpretation tree search software.

Another difficulty in pose estimation was observed during these experiments. It is important that at least three surfaces which, in some sense, effectively span three-dimensional space, be present in the sensed image. By "effective spanning," we mean that three planes which are almost orthogonal should be present in the image. Orientation estimates are unreliable, for instance, if only one plane's surface normal has a z component and if the x and y components are much larger than the z component for that plane. One example of this is the image in Figure 6.10. Our pose estimate for the object in this image was not as accurate as for the objects in Figures 6.11 and 6.12. The overlaid model wireframe is shown in Figure 6.13.

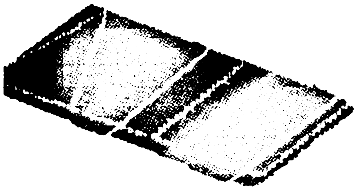


Figure 6.13. Wireframe of transformed model overlaid on sensed range image of Figure 6.10.

Parallel Implementation of IT Search

To investigate the feasibility of conducting the IT search on parallel machines, a version of interpretation tree search restricted to polyhedral objects was implemented on a serial machine, a Sun SPARCstation, and on two parallel MIMD machines, a 96 node BBN Butterfly GP-1000 at Michigan State University and a 48 node BBN Butterfly TC-2000 at Argonne National Laboratory. The GP-1000 has 96 MC68020 processing elements, each of which has 4MB of memory that can be accessed by all of the other processing elements over an interconnection network. The TC-2000 has a similar architecture, although it uses 48 MC88000 processing elements each with 16MB of memory. Programs written using BBN's Uniform System for parallel programming are portable between these two machines.

Interpretation tree search seems well-suited for implementation on these MIMD machines because nodes of the tree can be mapped onto different processors, each of which has enough power and memory to independently evaluate the pruning constraints on the relevant data. Two algorithms, identical except for having different levels of granularity, were implemented on the Butterflies. The programs consist of approximately 2000 lines of code. For timing comparison purposes, the same algorithm was implemented on the Sun SPARCstation.

All timings and speedups reported here for the Butterflies were found using the Uniform System call to the routine TimeTestFull(). Only the interpretation tree search was timed; the time taken to input the images and to build the internal models of the objects was not included. The time to initialize the Uniform System was also not included in the timings. The speedups reported are averages over several (usually about 10) trials. It was necessary to compute averages using "trimmed" means for configurations using large numbers of processing elements, because occasionally one trial exhibited much longer execution times than the other trials. This behavior appears to have been a consqeuence of sharing the machine with other users. Thus, we considered that trial an outlier and did not use it in calculation of the mean.

"Trimmed" means were only necessary for configurations that contained a large number of processors. For instance, a typical tree search can be executed in approximately 200 milliseconds in large-processor configurations. However, other Butterfly users and system applications cause contention for the Butterfly switch and I/O resources, thus inducing artificial delay into our timings. A small delay of only 50 additional milliseconds, for example, would reduce the apparent speedup of a 200 millisecond run by one-fourth.

The coarse-grained implementation achieved relative speedups of 4 or less while the medium-grained implementation achieved relative speedups of 9 to 14 for the objects tested.

Images, Models, and Feature Extraction

To test the feasibility of the parallel-processing approach, we conducted experiments on several images of three polyhedral objects. Our IT search on serial machines can also handle objects that have spherical, cylindrical, and conical surfaces, and we could easily extend those capabilities to our software on the BBN. However, in this chapter, the parallel search performance is compared to a serial search that has no capabilities for objects with curved surfaces. The three objects tested here are labeled Block1, Block2, and Block3. A range image of Block1 is displayed as a pseudo-intensity image in Figure 6.14[§] A Block2 image is displayed using a similar rendering in Figure 6.16. The other images used in our experiments are rendered in a "catalog" in Figures 6.18 to 6.23. All of the images are real range images except for the Block3 image which is a synthetically generated range image.

The segmentation of the range images and extraction of resulting surface features were done off-line on a SUN SPARCstation using the technique of Flynn [55], Hoffman, and Jain [75] discussed in Chapter 5. Surface segmentations of the two range images in Figures 6.14 and 6.16 are shown in Figures 6.15 and 6.17, respectively. Various features of the surface patches were then extracted and downloaded to the Butterfly.

The CAD models of the Block1, Block2, and Block3 objects, built by Flynn [55] using the IDEAS CAD package on a Sun workstation, were used for these experiments.

Two Parallel Implementations

Coarse-Grained Parallelism

Our first experiment was to test a coarse-grain parallelism, where one task was generated for each node at the first level of the interpretation tree. Each task performed a recursive depth-first search on its own subtree.

Speedup graphs for this approach applied to the polyhedral object called Block2 are shown in Figure 6.24. The speedups shown are from timings on the BBN GP-1000. Similar results were achieved on the TC-2000, although the speedups were never quite as high on that machine. The total number of hypotheses generated depended on the specific range image and varied from 12 to 229 for the five Block2 test images. The transformation arising from the correct matching of the surfaces in Figure 6.16 to model surfaces is shown in Figure 6.25. Clearly, the two viewpoints of the Block2 images are nearly identical.

[§] As noted earlier, this rendering displays brighter gray levels where object surfaces are oriented perpendicular to the viewing angle and darker gray levels for points on surfaces that are oriented parallel to the viewing angle.

Fi
ag

Figu
age 5

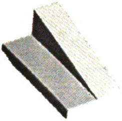


Figure 6.14. Pseudo-intensity rendering of Block1 Image 1.

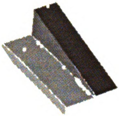


Figure 6.15. Segmentation of range image in Figure 6.14.

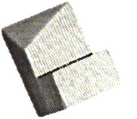


Figure 6.16. Pseudo-intensity rendering of Block2 Image 1.

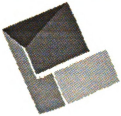


Figure 6.17. Segmentation of range image in Figure 6.16.

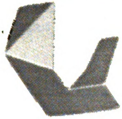


Figure 6.18. Block2 Image 2.

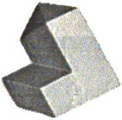


Figure 6.19. Block2 Image 3.

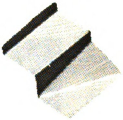


Figure 6.20. Block2 Image 4.

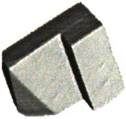


Figure 6.21. Block2 Image 5.

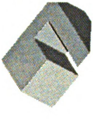


Figure 6.22. Block3 Image 1.

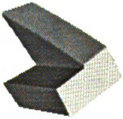


Figure 6.23. Block1 Image 2.

Figu
ima

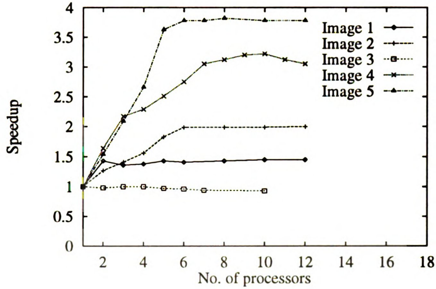


Figure 6.24. Speedups for coarse-grain parallelism for Interpretation Tree **Search for Block2 images.**

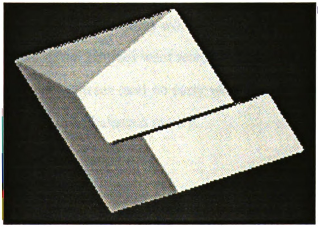


Figure 6.25. Transformation applied to Block2 model to match its image shown in Figure 6.16.

The spee
than 4 times
processors
running, an
greater than
be generate
powerful o

A diag
elements
activity in
BBN utili
(the trace
gist degra
others we
assigned
subtrees.
tree was
a deeper
The dee

Medium

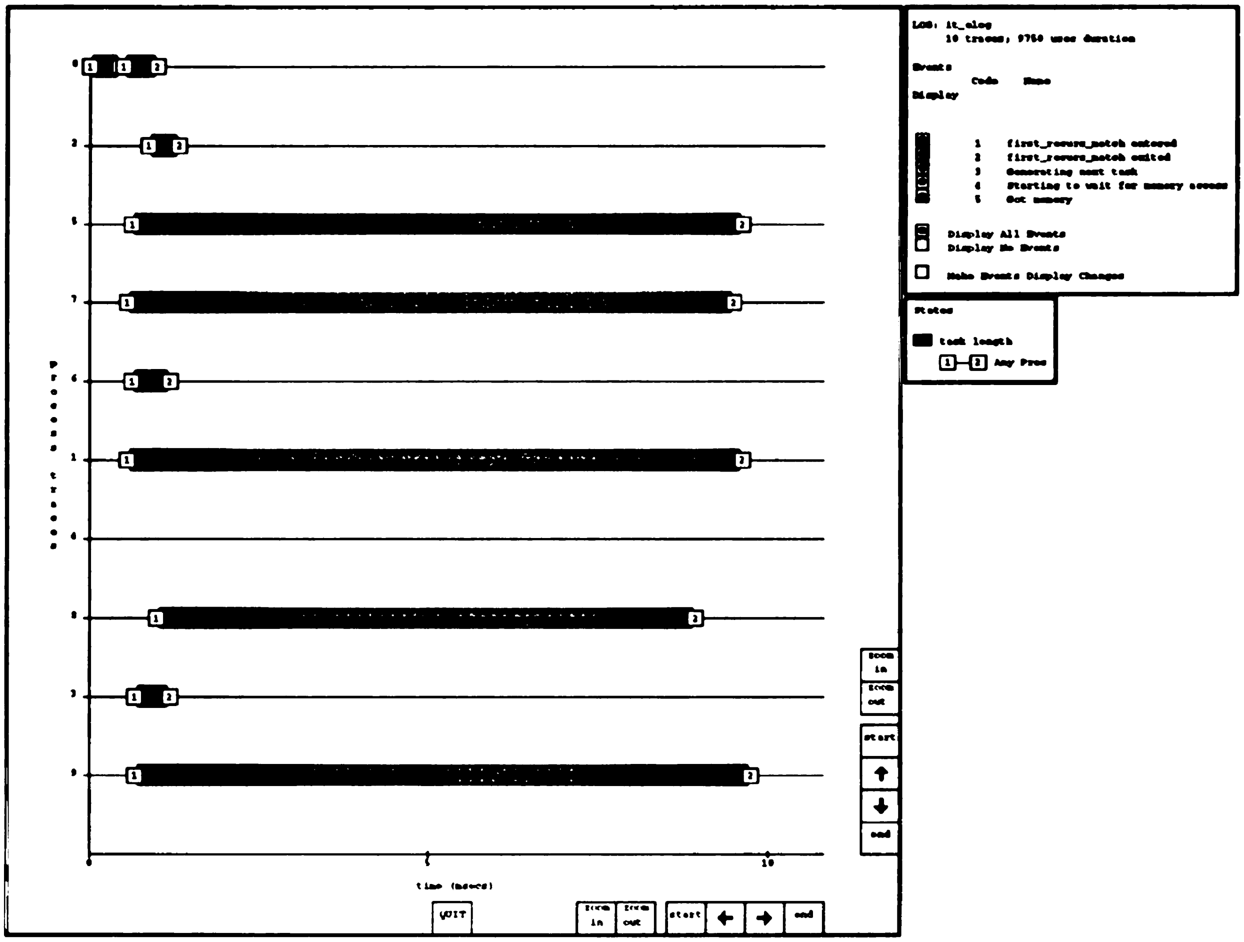
The inte
tasks in
parent
fashion
1, 2, …
mapped

The speedups observed were acceptable, but not impressive—we never achieved more than 4 times speedup. This is because early pruning of the tree resulted in most of the processors running out of work relatively quickly. A few of the processors continued running, and generated the majority of the matchings. In fact, the speedup can never be greater than $n + 1$, where n is the number of model surfaces, since only $n + 1$ tasks will be generated. The sublinear behavior may indicate that the pruning constraints are very powerful or that very strong evidence exists for the presence of a few of the model surfaces.

A diagram of the behavior of each processing element in a cluster of ten processing elements working on Block2's Image 5 is shown in Figure 6.26. This figure shows the activity in each processing element over time. The process traces were created using a BBN utility called *gist* and are a reasonable approximation of the behavior of the program (the traces are not a completely faithful portrayal because the bookkeeping necessary for gist degrades performance slightly). Five of the processors were utilized heavily while the others were grossly underutilized. Early violation of the pruning constraints in the subtrees assigned to the underutilized processors resulted in quick termination of the search of their subtrees. One processor was not assigned any work at all because the branching factor of the tree was less than 10. Five of the subtrees were searched for much longer times, indicating a deeper search within those subtrees (and no early violation of the pruning constraints). The deeper searches seemed to be balanced evenly.

Medium-Grain Parallelism

The interpretation tree search was parallelized at a medium-grain granularity by generating tasks in a top-down fashion. That is, a task is generated for a child node only after the parent node had satisfied the constraints. Since the tasks were generated in a top-down fashion, surface matchings were only attempted for image surface i after image surfaces $1, 2, ..., i - 1$ had been matched to some model surfaces. Ideally, children would each be mapped onto a different processing element. Figure 6.27 shows the speedup graphs for



LOG: it_alog
10 traces; 9750 usec duration

Events
Code Name
Display

1 first_recurs_match entered
2 first_recurs_match exited
3 Generating next task
4 Starting to wait for memory access
5 Got memory

■ Display All Events
□ Display No Events

□ Make Events Display Changes

States

■ task length
1—2 Any Proc

gist-5-tap-10nodes-hist.mwd
Thu Jan 9 15:17:25 1992

Figure 6.26. Process traces of coarse-grained IT search on a cluster of 10 processing elements applied to Block2 Image 5.

i

th

im

Im

Blo

surr

segr

inter

grapl

20 pr

the im

percer

Pro

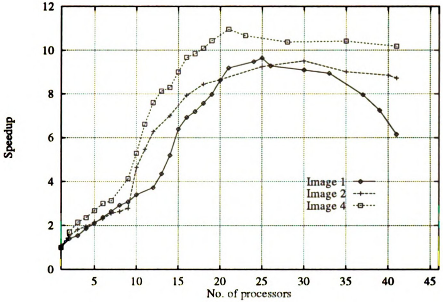


Figure 6.27. Speedups for medium-grain parallelism for Interpretation Tree Search on three images of Block2.

three images of the Block2 polyhedral object.

Figure 6.28 shows the speedup graphs for the Block1 images, the other two Block2 images, and for the Block3 image. Twenty-four hypotheses were generated for Block1 Image 1 and six hypotheses were generated for Block1 Image 2. Block2 Image 5 and Block3 Image 1 achieved the greatest speedups, probably because they contained more surfaces than the other images. (In fact, Block2 Image 5 was over-segmented by the segmentation scheme.) The large number of surfaces resulted in a deeper and wider interpretation tree, thus creating more tasks for the team of processors. All of the speedup graphs exhibit slightly sublinear behavior, although the region in the graph between 10 and 20 processing nodes is approximately linear. These results do not reflect any fine-tuning of the implementation. Through more careful coding, speedup could perhaps be ten to twenty percent higher.

Processor traces for the Medium-Grain parallelism implementation on an 8-node cluster f

b

th

fa

re

ve

im

gra

app

fact

the

by a

one

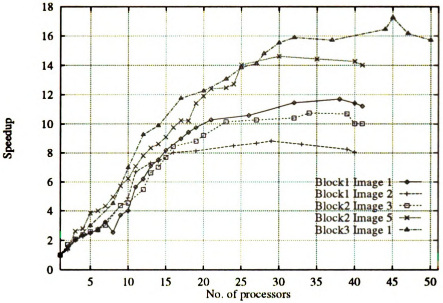
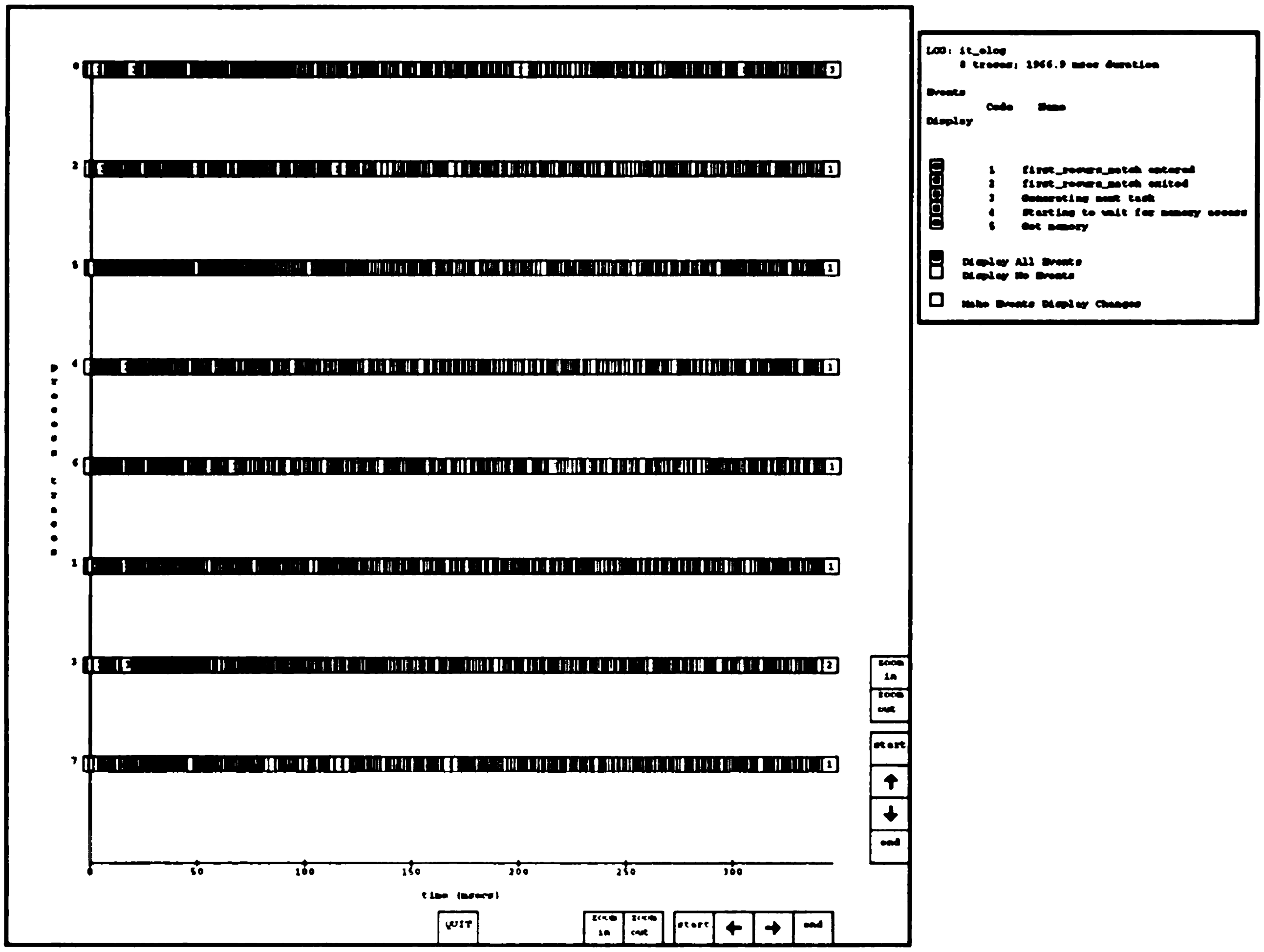


Figure 6.28. Speedups for medium-grain parallelism for IT Search on Block1, Block2 Images 3 and 5, and on Block3 Image 1.

for Block 2 Image 5 are shown in Figures 6.29 and 6.30. Figure 6.29 shows the trace at the beginning of execution while Figure 6.30 shows the trace at the end of execution. Although this output may appear to show that the coarse-grained implementation executed much faster than the medium-grained implementation, this was not the case. The instrumentation required to generate the gist output made the medium-grained implementation execute very slowly. Without instrumentation, the cluster of 8 nodes running the medium-grained implementation always executed much faster than 8 (or more) nodes running the coarse-grained implementation. Near the end of execution, several of the processing elements appear to be starved. This is probably because at lower levels of the tree, the branching factor is very small, commonly only 1 or 2. When a parent process creates child processes, the BBN Uniform System usually tries to minimize the number of immediate task switches by allowing the processor allocated to the parent process to immediately begin working on one of the child processes. Thus, at the bottom of the search tree, the node assigned to the

Fig
of 8



LOG: it_elog
8 traces; 1966.9 msec duration

Events
Code Name
Display

☑	1	first_recurs_match entered
☑	2	first_recurs_match exited
☑	3	Generating next task
☑	4	Starting to wait for memory access
☑	5	Got memory

■ Display All Events
□ Display No Events

□ Make Events Display Changes

gist-5-8nodes-start.ps
Thu Jan 9 15:32:17 1992

Figure 6.29. Process traces at the commencement of medium-grained IT search on a cluster of 8 processing elements applied to Block2 Image 5.

parent processor continues to be busy, generating little work for other processing elements. A graph of the processor utilization for the medium-grained search on the cluster of 8 nodes applied to Block2 Image 5 is shown in Figure 6.31. A histogram showing the distribution of execution times for each node of the interpretation tree is shown in Figure 6.32. A total of 6618 nodes of the tree were visited. This histogram shows that the tasks (nodes) were very evenly balanced across the processing elements, underscoring the fitness of IT search for parallel implementation.

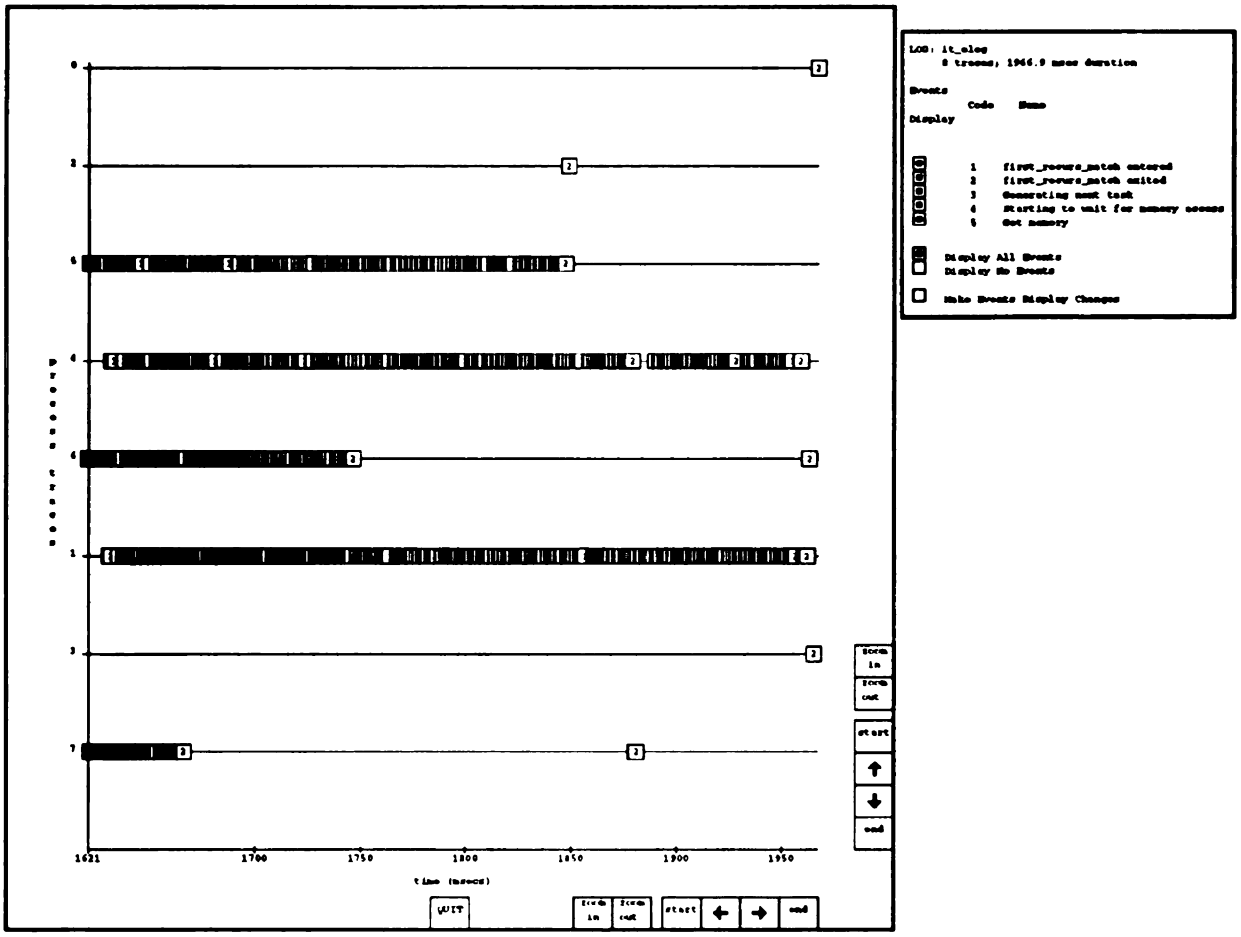
As processor configurations become larger, the task execution time increases as a result of increased memory contention and more overhead due to task creation. For example, in the 8-processor configuration for Block2 Image 5, average task length was 1.59 msec. For a 10 node configuration, average time was 2.1 msec. For a 30-node configuration, average time was 2.7 msec. These tasks are close to the ideal task length for the BBN GP-1000 [9], although performance would probably improve if task lengths were slightly longer.

Gustaffson [67] has suggested that speedups should be measured using a bounded speedup, in which the total amount of time, T, a process takes to execute is kept constant as the number of processors increases. Speedup is then measured by comparing the problem size which can be solved in time T using one processor versus the problem size which can be solved in time T using N processors. We could scale up the amount of work for our system by using models and/or images which have more surfaces, although this would involve measuring the amount of work done by the system as the number of processors is increased.

Graphs of the execution times on the GP-1000 are shown in Figure 6.33. Block2 Images 1, 2, 3, and 4 exhibited minimum times of between 0.08 and 0.23 seconds. Image 5 took a minimum of 0.95 seconds to execute.

Another set of experiments was conducted that consisted of executing the search and verification without producing any output. The speedups were roughly the same as those observed above, although the execution times were reduced by at least twenty percent. For

Figu
8 pr



LOG: it_elog
8 traces, 1966.9 msec duration

Events
Code Name
Display

☑ 1 first_recurs_match entered
☑ 2 first_recurs_match exited
☑ 3 Generating next task
☑ 4 Starting to wait for memory access
☑ 5 Got memory

☑ Display All Events
□ Display No Events

□ Make Events Display Changes

gist-5-8nodes-end.ps
Thu Jan 9 15:31:22 1992

Figure 6.30. Process traces at the conclusion of Medium-Grained IT search on a cluster of 8 processing elements applied to Block2 Image 5.

Fi
ele

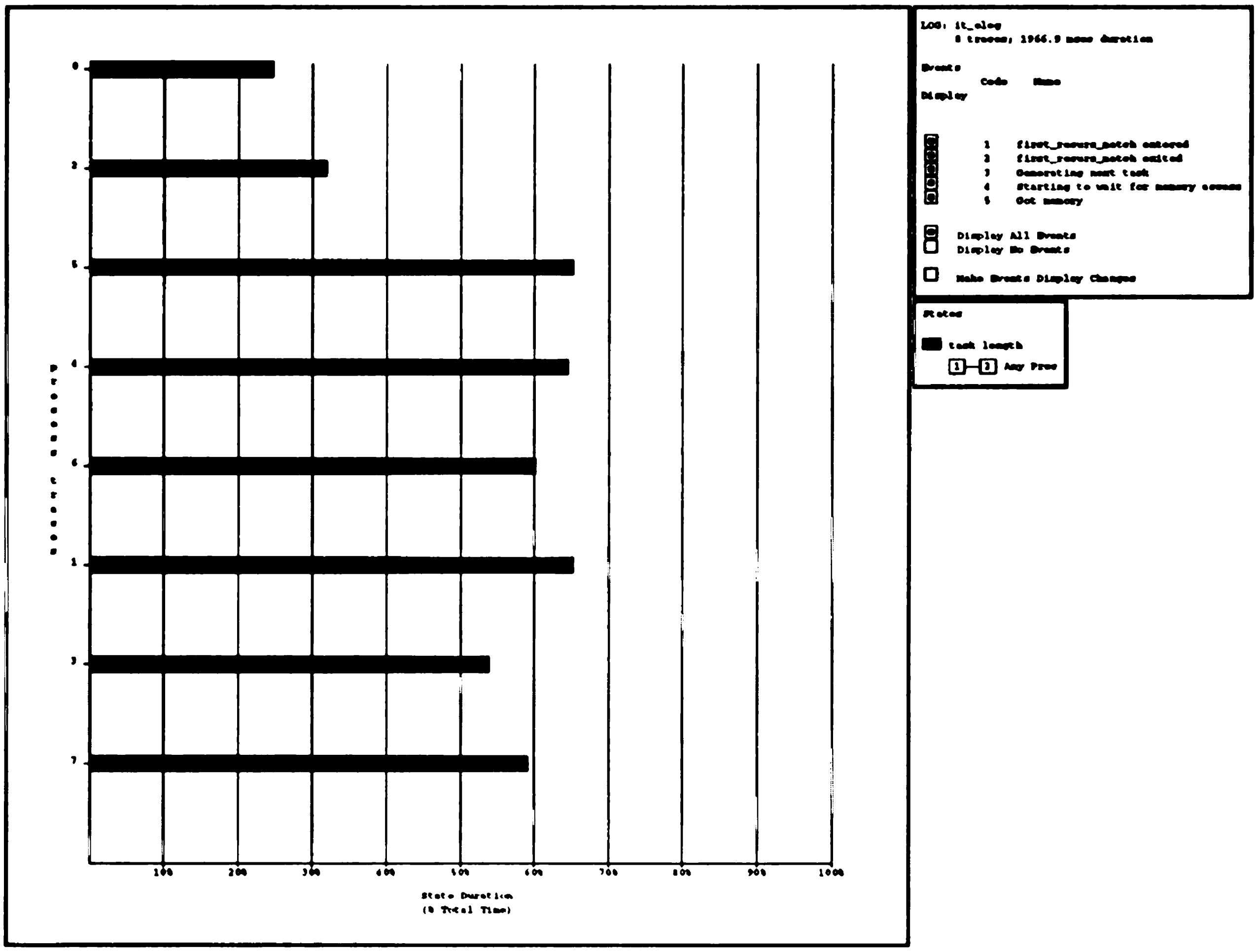
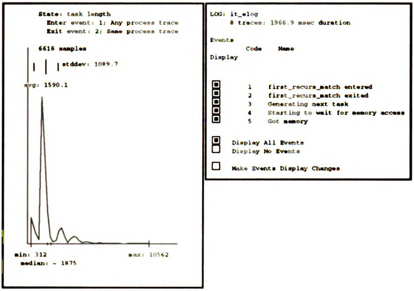


Figure 6.31. Processor utilization in medium-grained IT search on a cluster of 8 processing elements applied to Block2 Image 5.



```
State: task length
  Enter event: 1; Any process trace
  Exit event: 2; Same process trace

6618 samples
| | |stddev: 1089.7
avg: 1590.1

min: 312                              max: 10562
 median: ~ 1875
```

```
LOG: it_elog
     8 traces: 1966.9 msec duration

Events
        Code    Name
Display

☒       1     first_recurs_match entered
☒       2     first_recurs_match exited
☒       3     Generating next task
☒       4     Starting to wait for memory access
☒       5     Got memory

☒    Display All Events
□    Display No Events

□    Make Events Display Changes
```

Figure 6.32. Histogram of task length for medium-grained IT search on a cluster of 8 processing elements applied to Block2 Image 5.

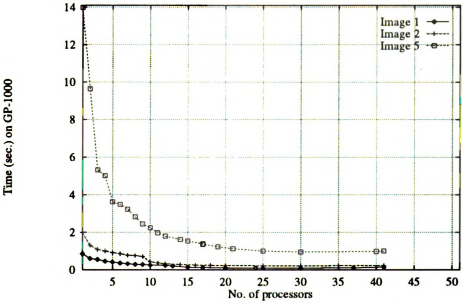


Figure 6.33. Time for medium-grain parallelism of IT search on Block2 Images 1, 2, and 5 on the GP-1000.

Machine	CPU Time
SPARC 1	4.05 s
SUN 4/390	2.84 s
SPARC 2	1.76 s
GP-1000	0.98 s
TC-2000	0.42 s

Table 6.1. CPU times for IT search for Block2 Image 5 on various architectures.

Block2 Image 3, execution time for a single processing element was reduced from 1.37 seconds to 0.43 seconds and the time using 12 processing elements was reduced from 0.25 seconds to 0.08 seconds.

For comparison, our algorithm was implemented on the Butterflies and on SPARCstation 1's and 2's. The chart in Table 6.1 compares the execution times for interpretation tree search to generate and prune the search tree and produce the model-to-view transformations on the various machines. The chart shows execution times for Block2 Image 5. The minimum time, 0.42s was achieved using 24 nodes of the TC-2000. The same program executed in 1.22 seconds using 20 nodes of the GP-1000 and in 0.98 seconds using 40 nodes of the GP-1000. The time for 40 nodes on the GP-1000 is the minimum time on it while the time for 24 nodes on the TC-2000 was smaller than the time for any other number of processing elements.

In general, the GP-1000 performed interpretation tree search approximately twice as fast as a SPARCstation 2. The TC-2000 was approximately 4 to 4.5 times faster than the SPARCstation 2. These execution times are still too large for many real-time inspection applications.

Several approaches were investigated to improve the performance of IT search on the BBNs. Most of the effort was concentrated on memory contention. We first tried to allocate the memory for child processes on the parent processor. Later, we allocated the

S

T

e

se

pa

sir

an

maj

time

memory at random. Allocating the memory on the parent processor produced slightly improved speedups, but only for configurations of a small number of processors. This is probably because one or more of the child processes were actually initiated on the parent's processor. In larger configurations, it is less likely for the parent node's processor to be selected to do the work. We also observed some improvement in performance for configurations of small numbers of processors if we used asynchronous task generation (while continuing to use synchronous task generation whenever the number of processors exceeded the branching factor at the top level of the tree). Synchronous task generation means that a parent node waits for all of its children to finish searching their subtrees before continuing any termination activities at the parent node (perhaps consisting of outputting information or updating data structures). Asynchronous task generation allows the parent node to continue processing while the children begin processing. In our experiment, we allowed the parent processing element to begin processing one of the child nodes. In small processor configurations, proper use of asynchronous task generation reduces some of the task-switching overhead by allowing the parent node's processing element to directly search one of its subtrees without operating system intervention.

Summary of IT Search

The IT search procedure seems well-suited to MIMD machines that have 10 to 20 processing elements. Parallel approaches to other high-level vision problems may have similar success.

Our experience in converting a serial implementation into a parallel one also presents several lessons. First, a simple conversion of code is usually not possible. At the minimum, parallel computation structures must be added to the code. If optimal performance is desired, a great deal of effort must be invested in reducing memory contention and choosing an optimal level of granularity. The non-deterministic nature of most parallel programs is a major barrier to program development, frequently producing quite unexpected—and sometimes unreproducible—patterns of execution. This complicates debugging and performance optimization. Typical parallel programming environments also complicate development. This is because development, debugging, and execution monitoring tools tend to be quite primitive at present (if they are available at all).

Future Directions for IT Search

In the future, several extensions to our interpretation tree search technique should be considered.

(i) The verification step is crude, and should use more of the sensed surface pixels to verify the hypotheses. Model and object surfaces should be compared directly, or at least a random sampling of points should be compared. We have used the techniques of Chapter 7 for verification of pose, and propose that those methods be used to evaluate the hypotheses that survive IT search's constraints.

(ii) The current search technique matches all object surfaces with a model surface. This usually results in too many hypotheses being evaluated. The approach by Marshall [111] (also suggested by Flynn [55]) of only matching enough surfaces to estimate an approximate model-to-view transformation, and using this as a constraint on matches deeper in the tree, should make the matching process faster, although the speedups on the parallel machines may not be as dramatic since there would be less work to distribute. It may not even be necessary to search to the bottom of the tree, although Marshall used the matches deeper in the tree to refine the original transformation estimate. Time performance and transformation accuracy should be explored for various search depths.

(iii) Different pruning constraints should be examined. Weaker constraints would result in a deeper and wider search tree that would make more work available for the processors. This might not be detrimental on the parallel machines because the processing elements are currently being underutilized. Stronger constraining features should also be examined; strong constraints will generate fewer and longer tasks but this could result in more efficient use of a small number of processors, leading to fast processing.

(iv) Our interpretation tree search on the serial SPARCstations already includes quadric surfaces and this functionality should be extended to our parallel implementation. The parameters of quadric surfaces could be estimated using the surface classification techniques discussed in Section 4.3. We could determine the rotational component of the model-to-view transformation for objects containing quadric surfaces using features such as the normal vector of the planar surfaces, the axis directions of cylinders and cones, and the shortest vectors between sphere center(s) and planes, cylinders, and cones, etc. The translational component of the transformation could be determined from examination of polyhedral vertices, sphere and cone centers, and the intersections of cylinder and cone axes with planar surfaces.

(v) The effects of memory contention should be explored further. The performance should be improved if memory contention can be reduced. Specifically, the effects of making local cache copies of data structures before referencing them should be investigated. Local cache copies can be created using the GP-1000 Mach Operating System's Do_bt() call. The effects of locking mechanisms and coding styles on memory contention should also be explored.

(vi) Finally, different granularities of implementation should be examined. A finer granularity could be achieved if parallelism within a task can be exploited. Perhaps a coarser parallelism would yield better results, however. Since there is a limited amount of work available near the bottom of the depth first search, it would seem that a finer granularity is in order, however.

Preliminary work on Interpretation Forest Search

If multiple objects need to be inspected, object recognition and localization are both prerequisites to inspection. Our parallel implementation of interpretation tree search as presented above can only "recognize" single objects. That is, interpretation tree search is useful for pose determination of a known object. A parallel interpretation forest (IF) search algorithm has also been implemented to allow the matching of scene objects to multiple models. We anticipated that this would enable the amount of work to be scaled-up and should deliver even greater speedups since the processors are currently underutilized, especially in clusters of twenty or more processing elements. In fact, our observed speedup measures for IT search may already be overly pessimistic because we have not scaled-up the amount of work as we scaled up the number of processors.

Our work in this area is preliminary. We have not investigated the effects of memory contention, task size, or number and type of models and objects on the performance.

We have tested the performance of coarse-grained and medium-grained implementations of IF search on both the TC-2000 and GP-1000. We were unable to achieve speedups for IF search that were as large as those observed for IT search. More study is needed to determine reasons for this. We hypothesize that memory contention becomes a much more serious factor in IF search, and believe that it is limiting the speedup. These effects should be investigated in depth.

The performance of a coarse-grained implementation of interpretation forest search is compared on the GP-1000 and TC-2000 for Block2 Image 5 with a model database of two models in Figure 6.34. In a one-processor configuration, the GP-1000 executed the search in 14.42 seconds while the TC-2000 executed in 5.8 seconds. With a cluster of 8 processors, the GP-1000 took 3.74 seconds and the TC-2000 took 1.43 seconds.

The medium-grained implementation for the same image and models took 3.45 seconds with 8 processors and 2.13 seconds with 18 processors on the GP-1000. (Since there is

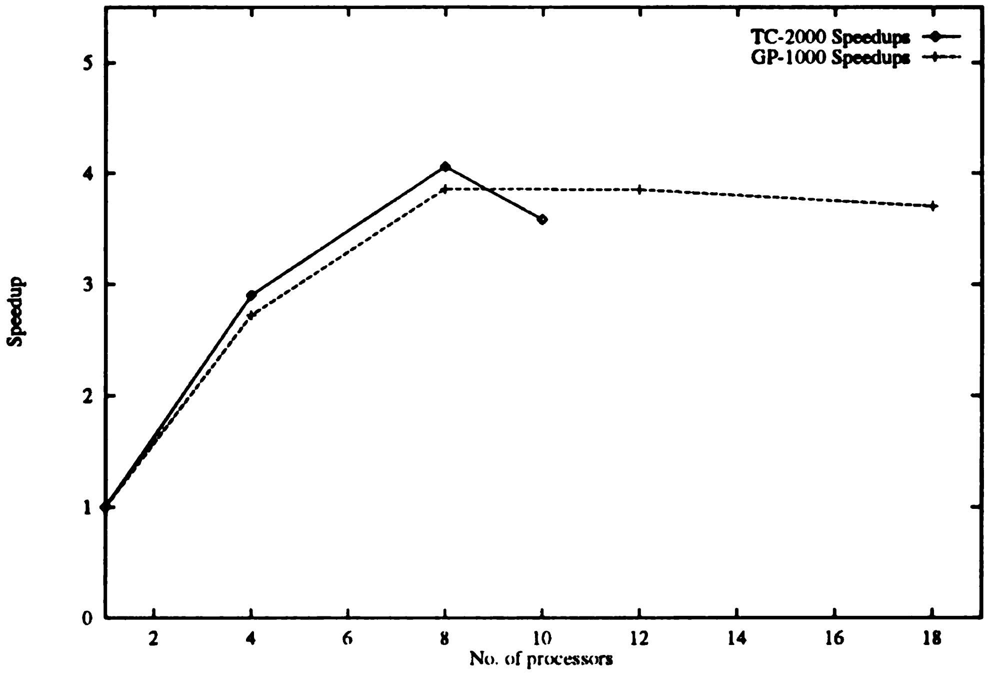


Figure 6.34. Speedups for coarse-grain parallelism of IF search on Block2 Image 5 on the GP-1000 and TC-2000. Search was for two models.

more work in the medium granularity implementation, better speedups could be achieved in configurations with a larger number of processors.) On the TC-2000, the medium-grained implementation took 1.1 seconds with 8 processing nodes and 1.02 seconds with 11 processing nodes. The speedup graph for the medium-granularity algorithm is shown in Figure 6.35.

If all three models are used in the recognition of Block2 Image 5, we observe slightly better speedups. On TC-2000, the coarse-grain parallelism exhibited a top speedup of 4.83 with 10 processors (time: 2.58 seconds). The coarse-grain speedup was 5.1 with 26 processors on the GP-1000 (time: 5.86 seconds). The medium-grain parallelism executed in 2.06 seconds with 10 TC-2000 processors and in 3.76 seconds with 28 GP-2000 processors (maximum speedups of 6.7 and 10.5, respectively). The speedups are plotted in Figure 6.36.

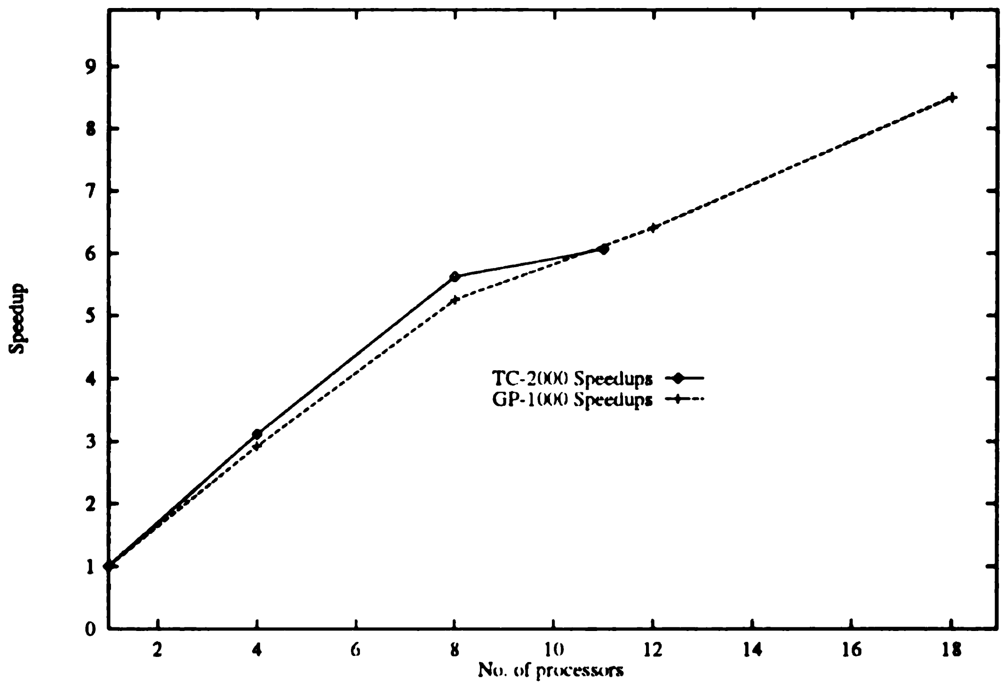


Figure 6.35. Speedups for medium-grain parallelism of IF search on Block2 Image 5 on the GP-1000 and TC-2000. Search was for two models.

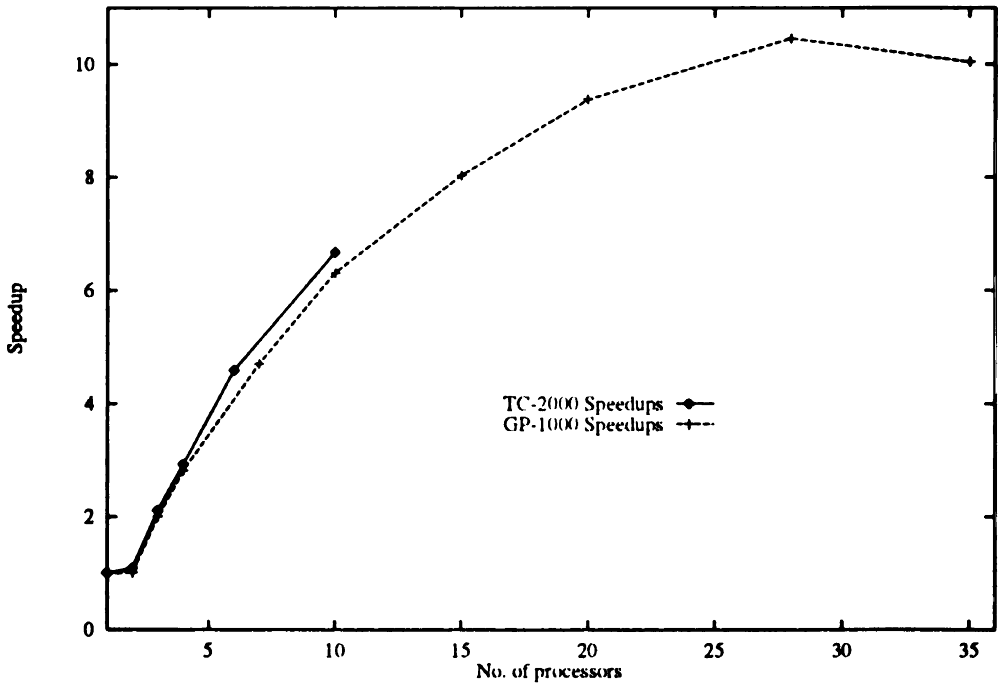


Figure 6.36. Speedups for medium-grain parallelism of IF search on Block2 Image 5 on the GP-1000 and TC-2000. Search was for three models.

6.5 Conclusions

In this chapter, methods for object localization have been presented. We discussed some of the popular techniques for localization in the computer vision literature and experimented with several techniques for use in pose determination for inspection.

We have used chamfering and the Hausdorff matching when there is only 2D rotational uncertainty (actually, chamfering is useful when there is 2D and limited 3D uncertainty). Both techniques seem to function acceptably within their 2D constraints. For 3D pose determination, we have tried to match images to stored views of a model. We also investigated the EGI method. Both techniques were problematic. Interpretation tree search seems to perform within acceptable limits for some vision tasks. We studied parallel interpretation tree search in detail in this dissertation and this is our proposed technique for 3D pose determination (and object recognition). At present, our interpretation tree search method may not be accurate enough for inspection purposes, however, and further work is required to enhance its performance in the presence of defects.

CHAPTER 7

Inspection of Gross Features

In this chapter, we present techniques for detection of gross defects. The techniques presented are useful for checking assembly integrity and for presence (or absence) of large features such as holes or slots. Ideally, objects would be inspected at several stages of the manufacturing process. For the casting inspection application, a part might be inspected first after it is cast, then again after a region of the part has been stamped by a press, again after a gate has been machined off the object, and finally after any other machining operations have been performed. The incremental inspection is useful because it limits the number of defective parts that continue through the manufacturing process.

Process inspection, which could be called incremental verification, can use information from previous verifications to simplify, or constrain, the current inspection operation. For example, the regions of the part that should not have changed since the last inspection could be tested quickly, although perhaps not completely. Instead, most of the inspection should focus on areas of interest in the object. An area of interest is defined to be any portion of the object that may have been altered, either accidentally or intentionally, by the assembly process.

A technique for fast incremental inspection is presented in this chapter. The entire object is first inspected for gross defects. Later, regions of interest can be inspected using the techniques of Chapters 8 and 9.

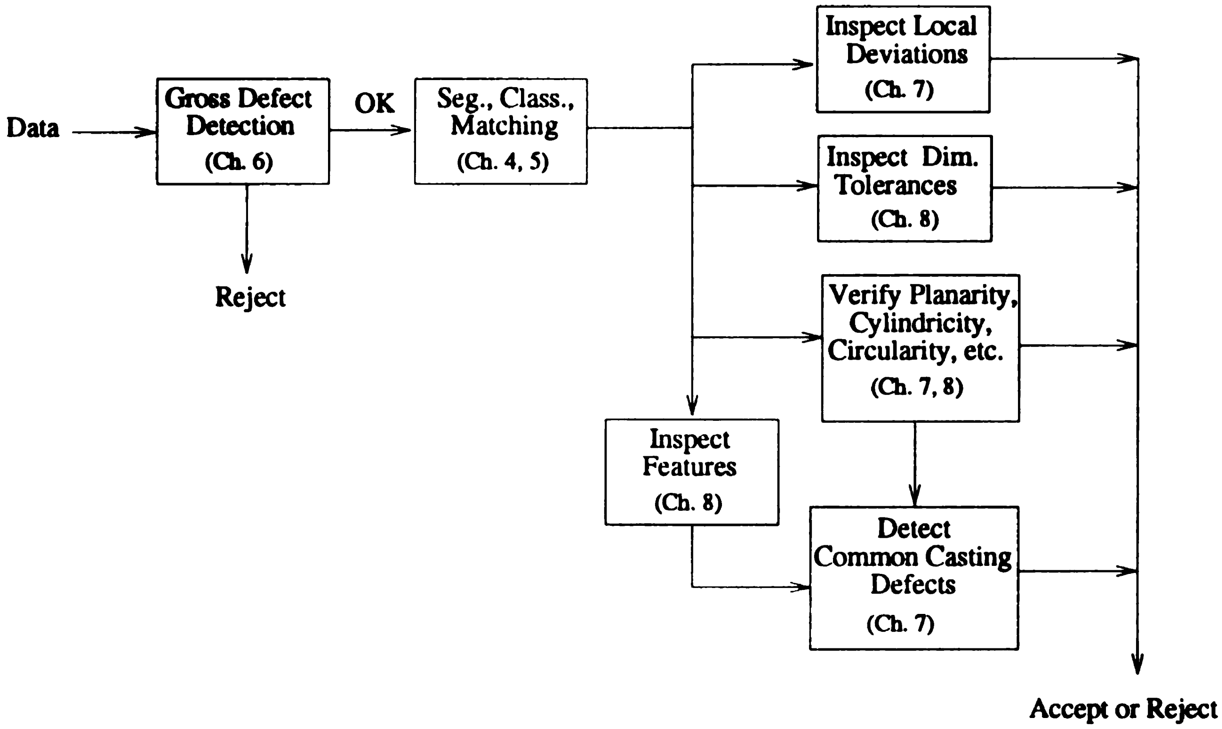

Data
Gross Defect Detection (Ch. 6)
OK
Reject
Seg., Class., Matching (Ch. 4, 5)
Inspect Local Deviations (Ch. 7)
Inspect Dim. Tolerances (Ch. 8)
Verify Planarity, Cylindricity, Circularity, etc. (Ch. 7, 8)
Inspect Features (Ch. 8)
Detect Common Casting Defects (Ch. 7)
Accept or Reject

Figure 7.1. Block diagram of steps in inspection system.

A block diagram of the steps of our inspection system is shown in Figure 7.1. The chapters that discuss each inspection step are also indicated within the boxes.

7.1 Random Subtemplate Matching

We perform in-process inspection by testing a random sampling of range pixels from an image synthesized from the CAD model of a defect-free object against the range pixels of an object image [128]. Random subtemplates have commonly been used as the first stage of a two-stage template matching procedure [105, 149]. Several template-based inspection systems have been presented in the literature and were discussed in Section 2.2.4. Template-matching has also been used by Bolles and Cain [16] in their local-feature-focus technique as a verification step.

Our verification method is efficient and is proposed as the first step of a two-stage verification process. Several parts, including castings and other small objects composed of

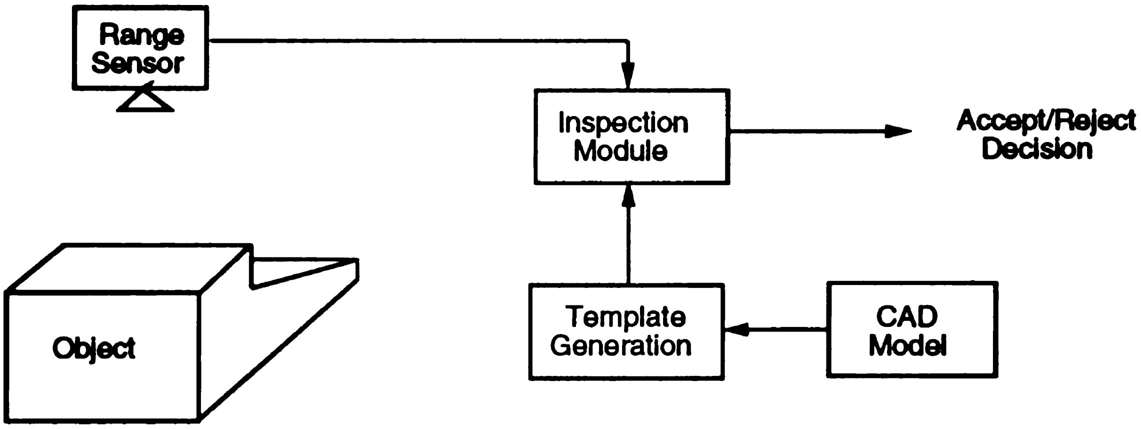


Figure 7.2. Block diagram of gross defect inspection using template matching.

polyhedral and/or quadric surfaces, were inspected for defects using this coarse verification technique.

This technique verifies object features against the features extracted from the CAD model. An outline of the inspection scheme follows. At each assembly step where verification is required, range data is collected from one (or more) viewpoint(s) to verify the part's integrity. A subtemplate, consisting of a small set of randomly sampled 3D points from an image synthesized from the CAD model, is compared against the sensed data. Missing or extraneous subparts are detected by discrepancies between measured and expected position of points in the templates. A block diagram of the system is shown in Figure 7.2.

A small subtemplate was used to allow quicker execution than a brute force comparison of every 3D point in the model and object. Exhaustive template matching can be quite time-consuming, as has been mentioned in Section 2.2.4. The subtemplate $\{S_i, i = 1, 2, \ldots, M\}$, where M is the number of pixels in the subtemplate, was constructed by randomly sampling approximately 1%* of the "relevant" pixels from a synthetic range image generated from the CAD model. Only those pixels which were on object surfaces and not near region

*We attempted a few trials using subtemplates with approximately 0.5%, 2%, and 3% of the pixels. 0.5% was found to yield much less reliable results while 2% and 3% yielded only slightly more reliable results.

boundaries are "relevant." (One of the matching statistics relies on estimation of surface normals, and since computation of surface normals near region boundaries is unreliable, we do not include those points in the subtemplate). The subtemplate is augmented with points from each visible surface of the model to ensure that each region contributes to the matching score.

The matching technique assumes that the expected pose of the part is known, either through fixturing or by a localization procedure. Since the range images are depth maps generated on an $x - y$ grid, only a small (5×5) neighborhood N_i needs to be searched to find the closest object point O_j to subtemplate point S_i. An Euclidean distance metric was used to select the pixel O_j closest to the point S_i.

The point coordinates and the corresponding surface normal vectors were used as features from which several matching statistics were computed to test the null hypothesis (H_0) against the alternate hypothesis (H_A), where

H_0 : the object is a defect-free instance of the model, and

H_A : the object is defective or not in the correct pose.

7.1.1 Matching Statistics

Three families of statistics were examined:

MSE - The mean squared error in position of object points O_j relative to subtemplate points S_i, defined as

$$\text{MSE} = \frac{1}{n} \sum_{\substack{i=1 \\ O_j \in N_i}}^{n} \| O_j - S_i \|^2,$$

where n, $n \leq M$, is the number of subtemplate pixels for which there is a corresponding object pixel O_j in the neighborhood N_i.

MDN - The mean difference in normal vectors to the tangent plane at corresponding points in object and model, defined as

$$\text{MDN} = \frac{1}{n}\sum_{i=1}^{n}\left|\cos^{-1}\left(\vec{n_{o_j}} \cdot \vec{n_{s_i}}\right)\right| ,$$

where $\vec{n_{o_j}}$ is the normal vector to the object surface at O_j, $\vec{n_{s_i}}$ is the normal vector to the model surface at S_i, and n, $n \leq M$, is the number of subtemplate pixels for which there is a corresponding object pixel O_j in the neighborhood N_i.

Ratio(*d*) - The ratio of the number of times object pixel O_j was greater than a distance d from the corresponding subtemplate pixel S_i, defined as

$$\text{Ratio}(d) = \frac{\sum_{i=1}^{M} C(i)}{M} , \text{ where } C(i) = \begin{cases} 0 & \text{if } \| O_j - S_i \| \leq d \\ 1 & \text{if } \| O_j - S_i \| > d \text{ or no } O_j \text{ exists.} \end{cases}$$

These statistics were used to test H_0 against H_A. H_0 was accepted if the statistic was less than a threshold value T. The three tests of hypotheses were:

If MSE $> T_{MSE}$, Reject H_0;
If MDN $> T_{MDN}$, Reject H_0;
If Ratio(d) $> T_{Ratio}$, Reject H_0.

If H_0 was rejected, the object could be rejected as defective, or an additional verification step could determine the nature of the defect. If H_0 is accepted then a more detailed verification step is initiated to check for small defects within regions of interest.

We wanted to accept H_0 in all cases where the object was in its expected pose and reject it in all cases where the object was rotated 5° or more or translated 0.25" or more. For smaller tolerances on rotation or translation, different thresholds would be used.

We first present thresholds that were determined empirically to deliver acceptable results for synthetic noisy range images. We then present thresholds for real range images.

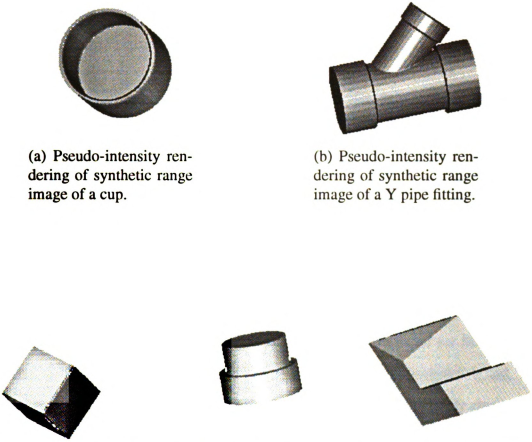
Performance

Generation of the random template, matching object pixels to the template, and testing the hypothesis using all the statistics took an aggregate time of less than one second on a SPARCstation 2. This time did not include the time to estimate the surface normals for the object, although that could probably be done at frame rates via convolution operation.

7.1.2 Synthetic Range Images

For synthetic images, the thresholds are as follows. The threshold value T_{Ratio} depends on distance d. Specifically, for d = 0.08", T_{Ratio} = 0.025; for d = 0.06", T_{Ratio} = 0.025; for d = 0.04", T_{Ratio} = 0.05; and for d = 0.03", T_{Ratio} = 0.1. The MSE statistic was tested at thresholds T_{MSE} of 0.001", 0.0015", and 0.002". T_{MDN} was tested for values of 3°, 2.5°, and 2°. These thresholds were determined empirically by examining the statistics for images where objects were moved by a small amount from their expected positions. In the experiments that follow, we will investigate the suitability of these statistics and thresholds for the inspection task.

In order to evaluate the significance level of the statistics, extensive Monte Carlo trials were conducted on synthetic noisy images generated from the CAD models using the polygon-conversion software discussed in Chapter 3. In these images, the object's true position and orientation was known. For each position or orientation to be verified, a set of 200 Monte Carlo trials (i.e., 200 images of an object) was conducted to estimate rejection rates for the statistics. Each trial consisted of testing one synthetically-generated noisy range image against the model. The noise was in the form of random Gaussian noise of standard deviation 0.01" added to the z component of each pixel.



(a) Pseudo-intensity rendering of synthetic range image of a cup.

(b) Pseudo-intensity rendering of synthetic range image of a Y pipe fitting.

(c) Pseudo-intensity rendering of synthetic range image of a cube.

(d) Pseudo-intensity rendering of synthetic range image of an adapter.

(e) Pseudo-intensity rendering of synthetic range image of Block2.

Figure 7.3. Range images from models of inspected objects.

Rot.	MDN		MSE			Ratio(d)			
	3.0	2.5	.002	.0015	.001	.08	.06	.04	.03
0°	0.0	.10	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1°	.31	.57	0.0	0.0	0.0	0.0	0.0	.26	.39
2°	1.0	1.0	0.0	0.0	0.0	0.0	0.0	.33	.99
3°	1.0	1.0	0.0	0.0	.08	0.0	.03	1.0	1.0
4°	1.0	1.0	0.0	.14	1.0	0.0	.50	1.0	1.0
5°	1.0	1.0	.48	.69	1.0	.48	1.0	1.0	1.0
10°	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 7.1. Rejection rates of H_0 for different object rotations; H_0: object is an instance of the adapter in an expected pose. Rates for 1° rotations are aggregate means from several sets of trials using different starting positions and rotations about different object coordinate axes.

Orientation Tests

Table 7.1 shows the rejection rates for each statistic in images of an adapter (see Figure 7.3(d)) at various rotations from an expected position. Note that the MDN statistic at the 2° threshold was omitted because it caused H_0 to be rejected for 98% of the 0° rotations. Similar behavior was observed for nearly all of the other objects tested. Thus, the 2° threshold does not appear to be useful for verification. This probably also indicates that our estimation of surface normals was not very accurate.

Figures 7.4 and 7.5 express the rejection rates for Block2 (Figure 7.3(e)) for the MDN, MSE, and Ratio(d) statistics at various thresholds. The MDN and Ratio(.03) were the most powerful statistics. The 3° threshold on the MDN statistic appears to offer good power and few false rejections. It is well-suited for small rotational tolerances of about 2°. The average distance statistic (MSE) also appears to be acceptable for verifying polyhedral objects like Block2. The Ratio(.08) statistic performed similarly to the MSE statistic at threshold 0.002 for the block. These statistics would be useful when the rotational tolerance for the block was 3°. The Ratio(.04) and Ratio(.06) statistics seem well-suited if the tolerance is about

Rotation	MDN		MSE			Ratio			
	3.0	2.5	.002	.0015	.001	.08	.06	.04	.03
0°	.03	.79	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1°	1.0	1.0	0.0	0.0	0.0	0.0	.063	.66	.667
2°	1.0	1.0	0.0	0.0	1.0	.005	.725	1.0	1.0
3°	1.0	1.0	.94	1.0	1.0	.99	1.0	1.0	1.0

Table 7.2. Rates of rejection of H_0 under various statistics; H_0: object is an instance of the cup. Results for 1° are aggregate means over 4 sets of 200 trials where each set of trials is a rotation in a different direction.

2°. The Ratio(.03) statistic is probably useful if the tolerance on rotation is very small.

In both of these tables, the results for 0.5° and 1° are aggregate averages over 4 sets of 200 trials where each set of trials is a rotation in a different direction. We also tested the rejection rate for rotations of Block2 from different starting orientations. Results similar to those in Figures 7.4 and 7.5 were observed.

Table 7.2 shows the rejection rates for incorrectly oriented instances of the cup (see Figure 7.3(a)). If the tolerances on rotation were about 2°, the MSE statistics or the Ratio(.08) statistics would appear to be appropriate for this object. The Type I error associated with using the MDN statistic at the 2.5° threshold is too large, suggesting that it may not be suitable for this object. The Type I error for the MDN statistic at the 3° threshold may also be too large. For small tolerances on rotation (i.e., approximately 1°) the MSE statistic at the .001 threshold and the Ratio(.04) and Ratio(.03) statistics may be useful.

A false rejection rate of 0.79 was observed for the MDN statistic at the 2.5° threshold for unrotated instances of the cup, however. This Type I error is too large, suggesting that the MDN statistic at threshold 2.5° is not suitable for that object.

The estimated rejection rates for incorrectly oriented instances of the pipe fitting (see Figure 7.3(b)) are shown in Table 7.3. The statistics exhibit similar rejection behavior although the Ratio(.06), Ratio(.04), Ratio(.03), and MSE statistic at threshold .001 may reject too many instances of very small rotations of the object. The other statistics are

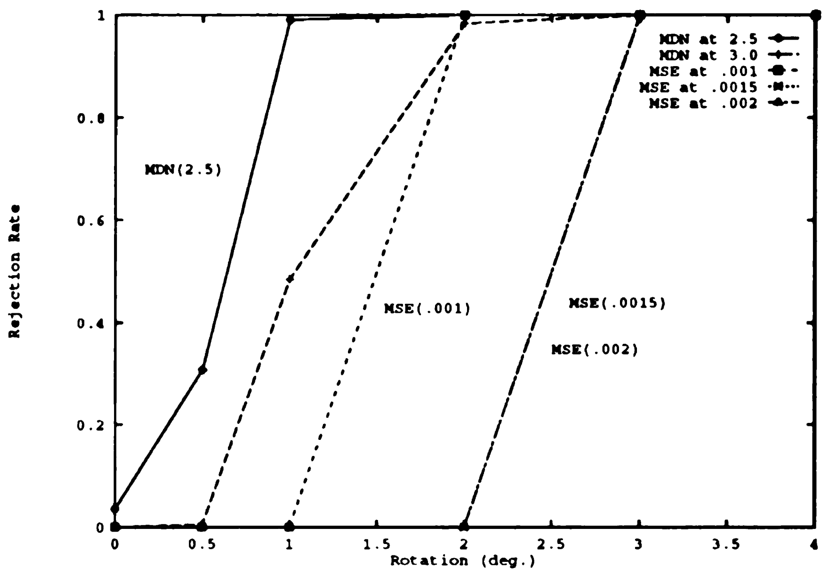


Figure 7.4. Estimated rates of rejection of H_0 under the MSE and MDN, and Ratio statistics with tolerances as shown; H_0: object is an instance of Block2. Rates for 0.5°, 1°, and 2° are aggregate averages over multiple sets of trials where each set of trials is a rotation along one of the object coordinate axis.

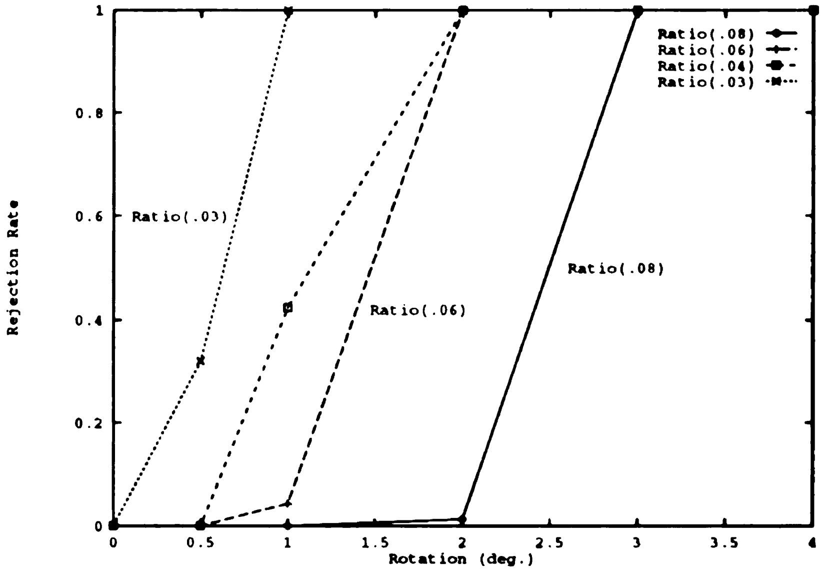


Figure 7.5. Estimated rates of rejection of H_0 under the Ratio(d) statistics with d as shown; H_0: object is an instance of Block2.

Rotation	MDN		MSE			Ratio			
	3.0	2.5	.002	.0015	.001	.08	.06	.04	.03
0°	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1°	0.0	.0075	0.0	0.0	0.5	0.0	.495	1.0	1.0
2°	.335	1.0	1.0	1.0	1.0	.515	1.0	1.0	1.0
3°	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 7.3. Rejection rates of H_0 under various statistics; H_0: object is an instance of the pipe fitting. Rates estimated using 200 Monte Carlo trials of synthetic range images of the pipe fitting with additive Gaussian noise of *s.d.* 0.01". Rates for 1° are aggregate means over 4 sets of 200 trials, with each set a rotation in a different direction.

Rotation	MDN		MSE			Ratio			
	3.0	2.5	.002	.0015	.001	.08	.06	.04	.03
0°	0.0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1°	.43	.853	0.0	0.0	0.0	0.0	0.0	.027	0.0
2°	.61	.995	0.0	0.0	0.0	0.0	0.0	.22	1.0
3°	.965	1.0	0.0	0.0	.49	0.0	.04	1.0	1.0
5°	1.0	1.0	.585	1.0	1.0	.995	1.0	1.0	1.0

Table 7.4. Rejection rates of H_0 under various statistics; H_0: object is an instance of the cube. Rates estimated using 200 Monte Carlo trials of synthetic range images of the pipe fitting with additive Gaussian noise of *s.d.* 0.01". Rates for 1° are aggregate means over 4 sets of 200 trials, with each set a rotation in a different direction.

useful when the rotational tolerance is about 1 degree.

The rates of rejection for various orientations of the cube are shown in Table 7.4. The MDN statistic at threshold 2.5° rejects H_0 for most of the rotated objects but also rejects some of the non-rotated objects. MDN at a threshold of 3° is better than at a threshold of 2°, but still is only useful when the rotational tolerance is small. The MSE, Ratio(.08), and Ratio(.06) statistics are useful when the rotational tolerance is approximately 3° to 5°. The other Ratio statistics are useful only when the rotational tolerance is quite small.

Position Tests

The rejection rates for translations of the objects by distances of 0.05 inches, 0.1 inches, 0.25 inches, 0.5 inches, and 1.0 inches in the x direction were also estimated. Results are presented here for translations of Block2. For translations of 0.25 inches or more, all of the statistics rejected H_0 for every trial, where H_0: the object is a defect-free, correctly positioned instance of Block2. The Ratio(.06), Ratio(.04), Ratio(.03), MSE, and MDN statistics rejected H_0 for every instance of 0.1 inch translations at the tested thresholds. The Ratio(.08) statistic accepted H_0 for all of the 0.1 inch translations, however.

The Ratio(.08) and Ratio(.06) statistics accepted H_0 for all of the translations of 0.05 inch. The Ratio(.04) and Ratio(.03) statistics rejected H_0 for all of the 0.05 inch translations, however. The MDN statistic at thresholds 2.5° and 3° rejected over 99% of the 0.05 inch translations. The MSE statistic at threshold .0002 and .0015 accepted all of the 0.05 inch translations but the MSE statistic at threshold .001 rejected 99% of the 0.05 inch translations.

Pose Testing Summary

Our tests indicated that the Ratio(.08) statistic and the MSE statistic at thresholds of .0015 or .002 appeared well-suited for verification of blocks when translational tolerances were approximately 0.1 inches and rotational tolerances were approximately 3° to 6°. The Ratio(.06) statistic was useful when the positional tolerances were 0.1 inches for translation and the rotational tolerances were about 2° to 4°. The other Ratio statistics were fairly useful when the rotational and translational tolerances were very small, although they may exhibit too many cases of false acceptances for rotated blocks to be very useful. The MSE statistic at the 0.001 threshold performed well when the rotational threshold was approximately 1° and the translational threshold was about 0.05 inches. The MDN statistic was fairly useful when the rotational tolerances were very small and the translational tolerance was about 0.05 inches.

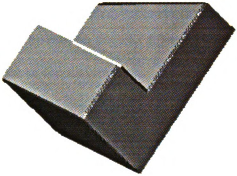


Figure 7.6. Block2 with a corner missing.

We obtained similar results in testing the cup, Y pipe fitting, and in testing instances of a cubic object. Through empirical testing of the performance of the statistics for a specific objects, a correct choice could be made for the desired inspection application.

Gross Defect Detection

Deformed objects were also generated for testing. The Block2 object with one of its surfaces deleted is shown in Figure 7.6. We conducted 200 Monte Carlo trials which compared images of this block to an ideal Block2 in the same position as the deformed block. H_0 was accepted by the MDN statistic at the 3° threshold for all of the instances of the deformed Block. The rejection rate for H_0 was 77% at the 2.5° threshold on MDN, however. The MSE statistic at all three thresholds and the Ratio(d), $d > 0.04$, statistics rejected H_0 for all the trials. Ratio(.03), though, accepted H_0 for all of the instances.

We also attempted verification of images containing a cubical object against the block model. The cube was positioned so its visible Y junction was coincident with the visible Y junction of the block. All of the statistics resulted in rejection of all of the cube images. It seems that all three families of statistics are well-suited for detecting large defects in the objects, even when substantial portions of the object closely match the model subtemplate.

A deformed block with 0.25 inches removed from one of the surfaces (with all the other surfaces being correct) was also tested. The MDN statistic with a 3° threshold accepted H_0 for all of the trials. At the 2.5° threshold, H_0 was rejected 4% of the time. All of the Ratio statistics accepted H_0 for each of the test images.

Our experimental results would seem to suggest that the MDN statistic has too high of a significance level to be useful for verification of many objects unless the tolerances on pose are quite small. In many of the tests, the MSE statistic had a very low power although it did seem to perform satisfactorily in cases where the object was positioned much differently than the model. Statistics such as Ratio(.04), Ratio(.06), and Ratio(.08) have strong discriminatory capability and seem to be the most worthwhile for use in this verification task. The Ratio statistics measure both the "closeness" of objects in two images and also the rate of local deviations from an ideal, so it is not surprising that they were the best-suited in these experiments. There may be some types of objects or defects that are detected more reliably by the other statistics, however.

7.1.3 Real Range Images

We applied the template-matching scheme to a large number of real range images of castings and other objects and discovered several limitations to the technique. The major limitation is sensor-related. Because of the large baseline between the camera and laser in the 100X sensor, self-occlusions are present in most range images. The occlusions tend to increase the value of the matching statistics (except MDN), making it difficult to distinguish defective parts from occluded parts. Another problem is that excess material on the object may not be detected. For example, Figure 7.7 contains the range image of a casting that has two defects including a blow hole and excess material in the center. Yet, the value of the matching statistics are very small for this image. Also, while the sensor depth resolution is quite good, the x and y resolutions are relatively poor (pixels are separated by a distance of 0.025" in

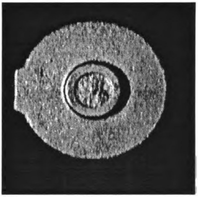


Figure 7.7. Image of a casting with two defects: blow hole and excess material obstructing a through-hole.

the x direction and by 0.04" in the y direction). Because it is unlikely for the sampling grid to be known in object and model images, this can lead to significant discrepancies during template matching. Finally, the template-matching technique is too inflexible, requiring rigid fixturing or precise localization.

To address these problems, the template-matching scheme was enhanced in several ways. First, the image from the CAD model was synthesized at a finer resolution than the object image. This minimized the effects of poor spatial resolution in the real range images. (Before, the pixel in the template might be as much as 0.02" distant from the closest pixel in the sensed image due to the sparseness of the depth map, thus yielding large scores for the Ratio(d), $d \leq 0.04$ and MSE statistics.)

To relax the positional requirements, we assume only that the object to be inspected is resting in a stable position on a known surface. For the gear blank casting, this would mean that it would be restricted to lying on its bottom planar surface on the assembly line, a position that would be likely even without the aid of fixturing devices. Positional uncertainty is resolved by aligning centers of mass of model and object images and then using the Hausdorff matching technique discussed in Chapter 6 to determine the rotation about the z axis to align the image silhouettes.

The problems of detecting excess material were resolved by using 2-way template matching (i.e., matching both from object to model and from model to object). Although both templates randomly draw 1% of the pixels from the images, since the model image has a higher resolution there are more pixels in the model template than in the template drawn from the sensed image. The matching scores were computed by aggregating the two matchings. For example, the new MSE statistic becomes

$$\mathrm{MSE} = \frac{1}{n_1 + n_2}\left(\sum_{i=1}^{n_1} \| O_j - S_{1i} \|^2 + \sum_{i=1}^{n_2} \| M_j - S_{2i} \|^2\right),$$

where n_1 is the number of model subtemplate pixels for which there is a corresponding object pixel O_j in the neighborhood N_i of model subtemplate pixel S_{1_i} and n_2 is the number of object subtemplate pixels for which there is a corresponding model pixel M_j in a neighborhood N_i of object subtemplate pixel S_{2_i}. The other statistics are extended similarly. Because the model template is larger than the object template, the model template has a stronger effect on the matching statistic than does the object template.

Finally, the effects of occlusion were reduced by acquiring range images from multiple known viewpoints and merging the data into a single image. We have used an additional viewpoint that is "behind" the object. In practice, this additional image could be acquired simultaneously with the original image taken from the "front" of the object by adding a second camera to the 100X sensing arrangement.

We tested these enhancements on several images of the gear blank castings. We performed one set of experiments on non-merged data and another set on merged data collected from two viewpoints. We found that the Ratio(.04) and Ratio(.03) statistics were most useful, although it was necessary to use thresholds different from those used for the synthetic images.

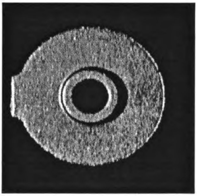


Figure 7.8. Image of defect-free gear blank 356 casting.

Statistic	Value
Ratio(.08)	.118
Ratio(.06)	.124
Ratio(.04)	.143
Ratio(.03)	.167
MDN	4.0°
MSE	.0009

Table 7.5. Matching statistic values for 2-way template matching for gross defect detection in the defect-free gear blank image.

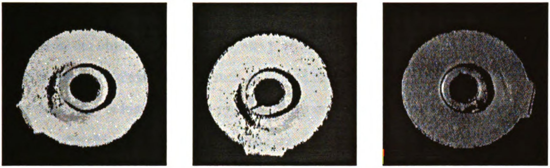
Application to a defect-free part

In the remainder of this section, we report the actual observed matching scores of the statistics rather than error rates at specific thresholds. In our first experiment, we tested the method on the image of a defect-free gear blank 356 casting shown in Figure 7.8. Table 7.5 shows the statistic values for 2-way template matching applied to the casting.

Applications to defective parts

Three images of a casting with a poured short defect were tested against an ideal defect-free image synthesized from the model. These three images are shown in Figure 7.9.

Table 7.6 shows the statistic values when a one-way template was used for the poured



(a) Pseudo-intensity rendering of Image 1.

(b) Pseudo-intensity rendering of Image 2.

(c) White Scanner Intensity plane for Image 3.

Figure 7.9. Range images of casting with poured short defect.

Statistic	Image 1	Image 2	Image 3
Ratio(.08)	.07	.074	.07
Ratio(.06)	.084	.08	.075
Ratio(.04)	.138	.142	.112
Ratio(.03)	.181	.20	.289
MDN	4.4°	4.6°	4.5°
MSE	.005	.002	.002

Table 7.6. Matching statistic values for 1-way template matching for gross defect detection in the 3 poured short images of Figure 7.9.

Statistic	Image 1	Image 2	Image 3
Ratio(.08)	.144	.14	.143
Ratio(.06)	.16	.155	.149
Ratio(.04)	.21	.21	.188
Ratio(.03)	.25	.264	.337
MDN	4.3°	4.6°	4.55°
MSE	.0046	.0020	.0016

Table 7.7. Matching statistic values for 2-way template matching for gross defect detection in the 3 poured short images of Figure 7.9.

short images. Table 7.7 shows the increase in values of matching statistics when using a two-way template. We would generally expect to see this increase mostly in cases of excess material in the object. The behavior was probably observed because the object has some small regions of excess material, such as around the gate, and because the template drawn from the model is biased toward regions of the model (namely, filleted regions) that are present in the object. If the model template was biased toward regions that were not present or were defective on the object, we would not observe this behavior. Specifically, since the model template is augmented with up to ten points from every surface, the model template tends to include many points from filleted regions. The cause of this are the bugs in the IDEAS CAD modeler's filleting feature. The filleting causes many polyhedral artifacts to appear along filleted surfaces, and the model template will be augmented to include up to ten points from each of these regions. The object template has no such bias. We could remove the bias from the model template, but then the template matching method would not be able to inspect for the presence of legitimate (non-artifact) features in the models. On this gear blank casting, there are many fillet artifacts, however. In fact, the rendering of the model at this orientation contains 45 regions, no more than 14 of which should actually exist in the model. Figure 7.10 shows the regions in the model for this view. Thus, the one-way template matching produces biased matching scores.

In the two-way template matching, thresholds of T_{Ratio} = 0.15 for $d = 0.04$ and T_{Ratio} = 0.2 for $d = 0.03$ seem to produce the desired discriminability, allowing the acceptance of defect-free parts and rejection of defective parts. It is not clear what threshold levels would be appropriate for the other statistics.

The two-way template matching for gross defect detection was also applied, first separately, to two images of a gear blank casting with a blow hole. These images, denoted A and B, are shown in Figures 7.11(a) and (b). The two images were later merged and the two-way template matching applied again. The defects in these images are fairly small and we do not expect the template-matching scheme to detect this type of error. The matching

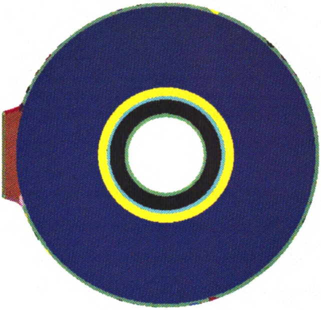
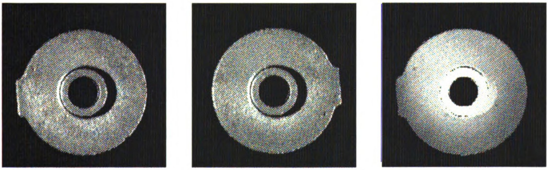


Figure 7.10. Labeling of surface regions for gear blank model exhibiting artifacts in filleted and rounded regions.



(a) Image A.

(b) Image B.

(c) Image of B merged with A.

Figure 7.11. Range images of gear blank 356 casting with blow hole defect. White scanner intensity map shown for (a) and (b). Depth map of merged image displayed in (c).

Statistic	Image A	Image B	Merged Image
Ratio(.08)	.13	.13	.12
Ratio(.06)	.133	.132	.12
Ratio(.04)	.14	.14	.123
Ratio(.03)	.166	.155	.128
MDN	4.3°	3.7°	3.9°
MSE	.0009	.0006	.00037

Table 7.8. Matching statistic values for 2-way template matching for gross defect detection in a gear blank 356 casting with blow hole defect (Figure 7.11).

scores for the two-way template matchings are shown in Table 7.8.

The two images were collected from known viewpoints, 180° apart. We merged the two images by applying the rotation transformation to image B and "filling in" the transformed values to all points in image A for which no range values were recorded. The translational transformation was computed by aligning the centers of mass of the two images. The merged range image is shown in Figure 7.11(c). Matching scores are presented in the last column of Table 7.8.

In these examples, the MDN and MSE statistics have not been very useful. The Ratio statistics seem better suited for detection of defects when position is controlled. The MDN and MSE statistics may be useful for some gross defects, although the defects would likely be much different than any of the most common defects encountered in the casting process.

7.1.4 Summary

We have presented a method for fast verification of given 3D objects in a known position and orientation. The verification was performed using a template composed of randomly selected pixels from a CAD-generated model that was matched to corresponding pixels in the object. Several matching statistics were defined to determine the goodness-of-fit between the template and object. The Ratio statistics, which were based on the number of object points sufficiently close to template points, offered the best discriminatory power in most of the cases tested. Merged data acquired from multiple viewpoints was used to reduce the effects of occlusion.

The coarse verification procedures assume that the object's pose is tightly controlled through fixtures. It is questionable if these techniques can be extended into a fixtureless environment. For instance, the techniques probably can not be extended to allow acceptance of objects that contain no defects but which are grossly misoriented since the methodology tightly couples detection of defects with detection of improper orientation. We propose this gross defect detection scheme for use in process inspection when the object is restricted to be in an expected pose.

7.1.5 Extensions

We propose that at the conclusion of the assembly sequence, or at important steps during assembly, images from multiple views could be collected to more thoroughly inspect the object. The approach presented in this chapter would allow quick incremental verification and complete final verification.

Our approach to coarse verification can also be used by a verification process after the object has been localized. Objects which pass the coarse inspection would be inspected to ensure their adherence to the dimensional tolerances and other features encoded in the model.

We could apply this verification technique to regions of interest in the object. We define a region of interest as any portion of the object that may have been altered, either accidentally or intentionally, by the assembly process. Our primary goal is to verify that operations of interest have succeeded. A dense template could be constructed for those regions and then compared to the object. Alternately, a pyramidal template could be used with additional dense verification performed as necessary in regions around pixels which do not match the coarse subtemplate well. Califano et al. [22] have suggested a similar multiresolution approach for visual perception that processes an entire range image at a coarse resolution and small windows of the image at finer resolutions.

Several other extensions should also be possible. After defect candidates are found, perhaps by considering object pixels which exhibit a poor correspondence to model pixels using the coarse defect detection procedures outlined above, the extent of the defect and the identity of the defect would be determined. Region growing around subtemplate pixels which are far from their expected positions can be used to determine defective regions. The defect region consists of all pixels adjacent to the original "bad" pixel; all neighboring pixels which are also far from their expected positions are added to the defect region. A related technique currently being investigated is determination of all pixels in the image which do not fall within a certain distance of their expected position. The first technique is probably quicker although it may not find all defective regions.

7.2 Conclusions

In this chapter, we have presented a template-matching scheme and matching statistics useful for detection of gross defects. First, a 1-way template matching scheme was proposed as a verification technique to determine if an object was positioned correctly. A 2-way template matching scheme for gross defect detection was then proposed and demonstrated on real data. The techniques perform well and execute quickly. The gross defect detection scheme is envisioned as a general in-process inspection step that would be used to quickly eliminate from consideration parts that contain gross defects. The matching statistics are probably also useful to evaluate pose hypotheses, perhaps in conjunction with one of the localization schemes of the previous chapter, such as IT search.

CHAPTER 8

Surface Shape Inspection

8.1 Introduction

Several classes of defects are commonly encountered in iron castings. Most of the defects could probably be classified into three categories—cracks, excess material, or insufficient material. In this chapter, the topic of detecting the presence of insufficient or excess material on planar or cylindrical surfaces of castings is addressed. Common examples of insufficient material include pits or holes, while a common example of excess material is a drop.

As indicated in Chapter 5, our surface classification algorithms [126] and [127] are used for fitting quadric (cones, cylinders, and spheres) or planar surfaces in segmented range images. Those techniques perform model-driven fittings to image surfaces. Poor surface fits probably indicate the presence of a surface shape defect on the object. It is possible that some surface defects such as mars, scratches, and irregular finish can be detected using these techniques, but, in general, those defects cannot be detected merely by examining errors in surface fit. In this chapter, we focus on inspecting surface shape, including planarity, cylindricity, and insufficient and excess material defects.

A rich body of literature exists for surface inspection. Many studies on image texture, for instance, are applicable for surface inspection. It is beyond the scope of this dissertation to explore surface inspection in detail. In fact, the complex mathematical presentations for surface inspection using texture often have little practical application in manufacturing. This is because defects like cracks and pits in metal surfaces can almost always be easily detected by edge detection under bright lighting. Instead, most of our effort focuses on surface shape inspection. We did attempt to find cracks in range images using the proposed techniques for insufficient and excess material detection but were unsuccessful. We were able to detect some cracks using edge operators on intensity images illuminated with carefully controlled directed lighting, however. In practice, complete inspection is probably possible only if multiple sensory modalities are used.

8.2 Pits and Excess Material Defects in Planar Surfaces

For the castings used in our investigations, the factory allowance for excess material is 0.120" or less deviation from the reference surface. The allowance for insufficient material is 0.060". Castings that have pits deeper than 0.060" are considered defective. The diameter of a pit is usually the same as its depth; thus, the diameter of a pit defect is usually 0.060" or greater. Surface deviations that are shallower than 0.06" are not of concern because the casting typically contains about 2mm (about 0.079") of finish stock. Thus, all material that surrounds pits of less than 0.060" depth will be removed from the casting during subsequent machining. However, small pits (i.e., pits whose diameters are 0.04" or less) of depth 0.03" or greater should also probably be rejected, as discussed in Chapter 1.

In this section, we discuss our technique for inspecting the most common casting defects, namely excess or insufficient material defects on planar surfaces.

8.2.1 Roughness of castings

Surfaces of unmachined castings are very rough. The roughness of the metallic surface is caused by the grain of the sand in the cope and the drag into which the metal is poured. There are also other marks on the castings, including scratches caused by rubbing against other castings and identification markings such as part numbers and logos. This lack of smoothness is a typical and difficult problem for machine vision inspection systems.

Measuring Surface Roughness

With the generous assistance of Dr. Salim Hiziroglu of Michigan State University's Forest Products Lab, a Bendix Microcorder depth gage (based on the stylus method) was used to obtain surface profiles from two 1" sections of planar surfaces on each of two gear blank 356 castings. These profiles provide reference information about the casting, allowing us to compare surface information acquired in range data with the actual surface deviations. The stylus profiling method evaluates the surfaces in two dimensions. A stylus is passed over the material in one direction and stylus position and vertical deviation are recorded. Graphs of one of the profiles from each of the two castings are shown in Figure 8.1. In the graph, vertical grid lines are 0.05" apart and horizontal grid lines are 0.001" apart.

Dr. Hiziroglu's equipment computed three standard roughness measures for ten intervals of each collection of profile data. These are (i) R_a, the arithmetic mean roughness value, (ii) R_{max}, the maximum peak-to-valley height, and (iii) the mean deviation of the surface from a reference height. The following expression defines R_a [*] [76]:

$$R_a = \frac{1}{l}\int_0^l |f(x) - \mu|\,dx,$$

where l is the evaluation length of the profile, $f(x)$ is the height of the profile at x, and μ is the mean value of the profile.

The values of R_a, R_{max}, and Mean Height (all values are in inches) for the profiles are shown in Table 8.1. Profiles 1A and 1B are from two different regions of one planar surface of one of the gear blank 356 castings. Profiles 2A and 2B are from the same surface on a different gear blank 356 casting.

[*] Based on DIN and ISO standards.

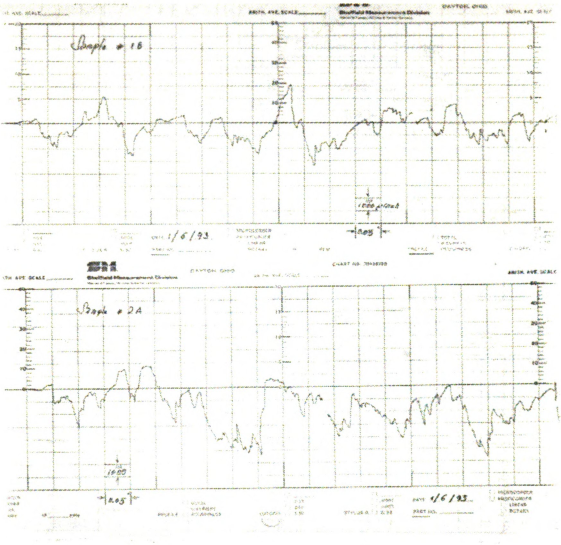


Figure 8.1. Surface profile graphs of planar casting surface.

Profile Number	Mean Height	R_a	R_{max}
1A	-0.001863	0.000498	0.004399
1B	-0.000371	0.000707	0.006307
2A	-0.001625	0.001052	0.007374
2B	-0.000426	0.000573	0.005436

Table 8.1. Measures of Surface Roughness.

Profile Number	Mean	Standard Deviation (σ)
1A	-0.0020	0.000829
1B	-0.0007	0.002539
2A	-0.0018	0.001558
2B	-0.0004	0.000865

Table 8.2. Estimated sample means and standard deviations of surface profiles.

Because we did not have access to the raw surface profile data, it was necessary to estimate the deviations from the graph. We estimated the mean and standard deviation of these surface profiles based on 19 samples from each graph. We did this at regular intervals of 0.05". Table 8.2 contains the estimated sample means and variances (all in inches). Ideally, the standard deviation σ of the surface could be derived from R_a, however there is no clear way to directly express σ in terms of R_a. It may be possible to bound σ by an expression involving R_a, however.

Sensed surface data

We also investigated the roughness of planar surfaces in castings using range images acquired by the Technical Arts White Scanner in the PRIP Laboratory. Figure 8.2 shows a range image of a planar patch (Patch 1). A plot of the density of the deviation of points from the plane (fitted using the method of Chapter 5) for the patch is shown in Figure 8.3. A histogram of the deviations for the patch is shown in Figure 8.4. The bounding box of this patch is approximately 1" by 2". The patch contains 2530 pixels. A plot of the density of the point deviations for a second patch is shown in Figure 8.5. This patch is bounded

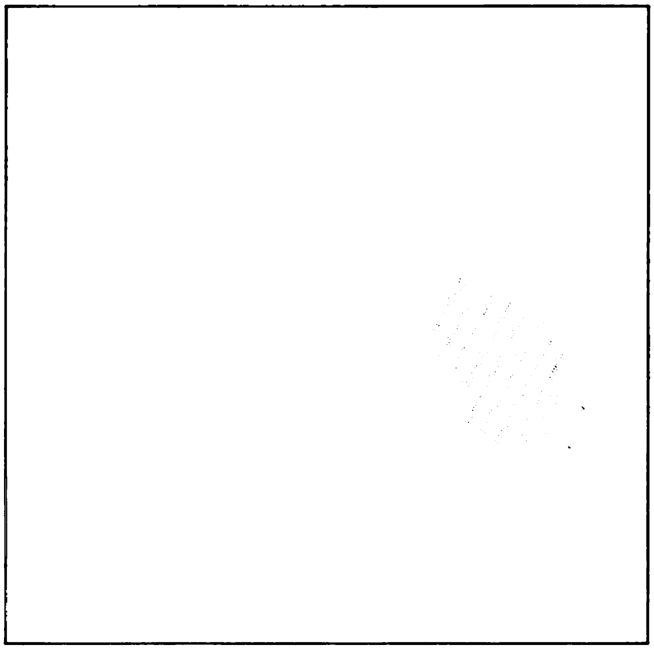


Figure 8.2. Defect-free planar patch (referred to as Patch 1) on casting.

by a box of size 1.7" by 1.2" and contains 1802 pixels. The deviation of the sensed points from these planar surfaces was approximately Gaussian.

A Chi-squared goodness-of-fit test for normality was applied to the data from Patch 1 (shown in Figure 8.2) and is shown in Table 8.3. The cutoff value for this data at the 0.05 significance level is 27.6 (assuming 17 degrees of freedom), so the assumption of normality cannot be rejected. The observed Chi squared value, 26.42, is very close to the cutoff value, however. In our experiments, we have, therefore, assumed that deviation of depth data from planar castings as sensed by the White Scanner is approximately Gaussian. However, there seems to be some departure from normality in the tails of the distribution. This may be related to physical limits in the manufacturing process, which places definitive limits on maximal local deviations, restricting the possibility of deviations larger than some amount.

We also performed a Chi-squared goodness-of-fit test on the data from Patch 2 and found that we could not reject the hypothesis of normality, again at the 0.05 significance level.

A normal Q-Q plot of the data from Patch 1 is shown in Figure 8.6. This plot shows

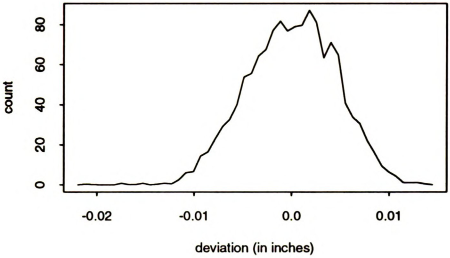


Figure 8.3. Density of deviations from a plane for surface Patch 1.

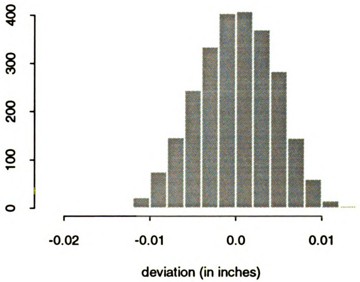


Figure 8.4. Histogram of deviations from plane for surface Patch 1.

Class	Deviation	Obs. Freq.	Exp. Freq.	χ^2 value
1	-0.00747	138	126.5	1.045
2	-0.00582	128	126.5	0.018
3	-0.00471	131	126.5	0.160
4	-0.00382	125	126.5	0.018
5	-0.00306	129	126.5	0.049
6	-0.00238	110	126.5	2.152
7	-0.00175	116	126.5	0.872
8	-0.001146	130	126.5	0.097
9	-0.00057	117	126.5	0.713
10	-0.000002	113	126.5	1.441
11	0.000574	118	126.5	0.571
12	0.001151	108	126.5	2.706
13	0.00175	136	126.5	0.713
14	0.00238	139	126,5	1.235
15	0.00307	133	126.5	0.334
16	0.00382	116	126.5	0.872
17	0.00471	162	126.5	9.962
18	0.00582	137	126.5	0.872
19	0.00747	134	126.5	0.445
20	0.014	110	126.5	2.152
Total				26.42

Table 8.3. Chi-squared goodness-of-fit on surface Patch 1.

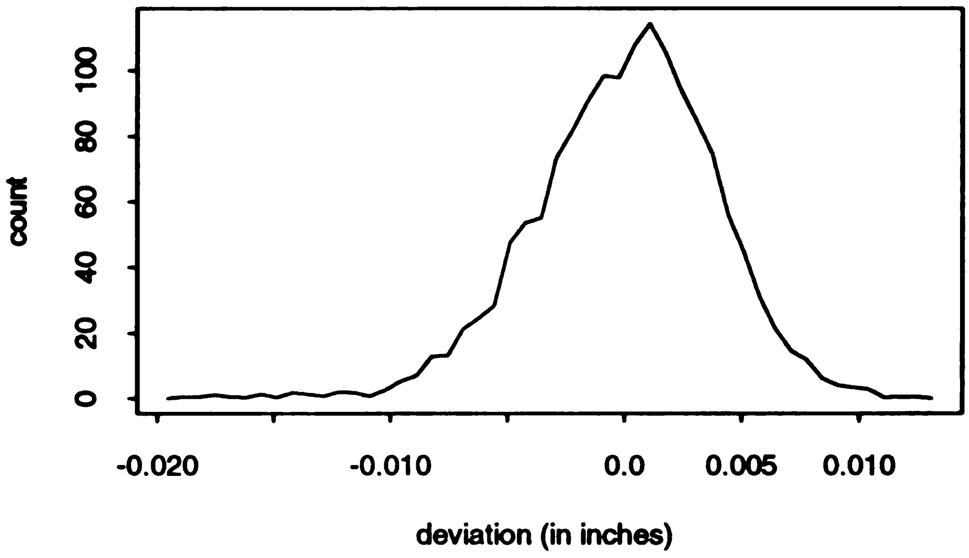


Figure 8.5. Density of deviations from plane for surface Patch 2.

the sorted Patch 1 depth data plotted against the quantiles of the normal distribution. The data would be considered to be normal if all the points fell approximately on a straight line. The upper and lower tails do appear to depart slightly from the straight line, but the points appear to be reasonably straight, so the assumption of normality is probably acceptable.

8.3 Method for Detection of Planar Defects

We define a defective pit or drop as having a surface area of at least two connected pixels in the sensed image whose depth deviates more than the allowable defect threshold. Unfortunately, many small pits (e.g., almost every pit of depth greater than 0.06" and many pits of diameter 0.04" or less) often are occluded from the view of the camera of the 100X range scanner. For such defects, the method for through-hole inspection and "void region" pit detection in Chapter 9 is usually sufficient, however. In detection of planar defects, we

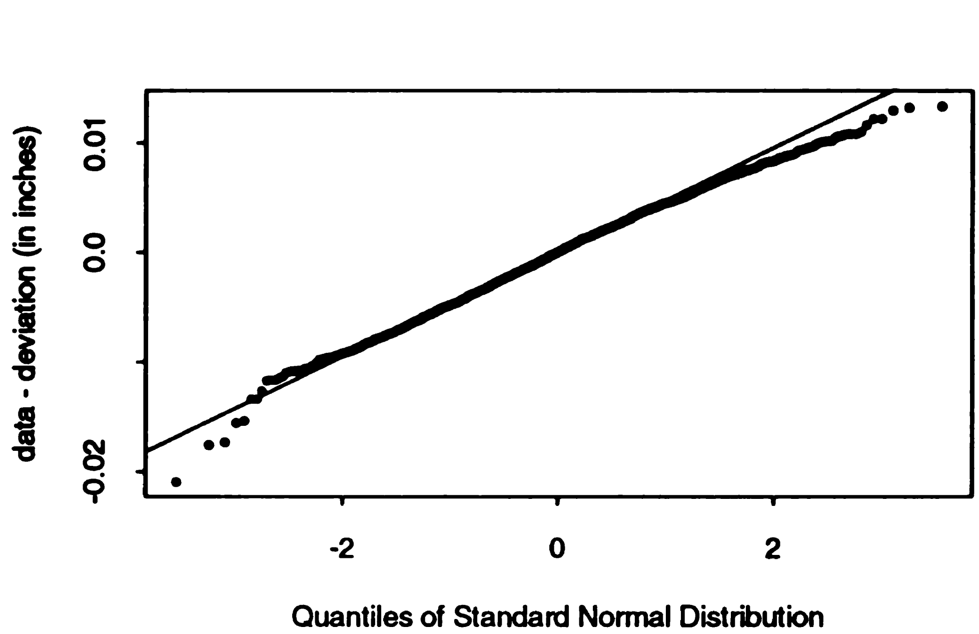


Figure 8.6. Normal Q-Q Plot of surface Patch 1.

also define as defective any local region where there is a single "void" pixel that is adjacent to a pixel whose depth deviates from the plane more than the planarity tolerance.

In the following discussion, we assume a fixed image size of 6.24" over 240 pixels in the x direction and 10.8" over 240 pixels in the y direction. This work envelope is typical for the 100X scanner. Thus, a pit or drop will be detected only if it is 0.05" or larger in diameter, consistent with the definition of insufficient material (actually, this caveat is more aggressive than the actual definition). To detect pits, we define a ***defect threshold*** of 0.025" deviation from the plane within local regions of the casting surface. We also define a ***planarity tolerance*** of 0.010" within the local region. We then iteratively fit local planes in the image. After the first iteration, the planes are fit to the set of points that lie within the planarity tolerance of the previously fit planes. (Points that do not meet the planarity tolerance from the previous plane could be defects or could be due to poor segmentation.) Pixels which deviate from the plane by more than the defect threshold of

0.025" are considered to be pits or other errors. A more global insufficient material defect can be detected by finding all points that deviate more than 0.06" or 0.12" (for insufficient and excess material, respectively) from a global fit.

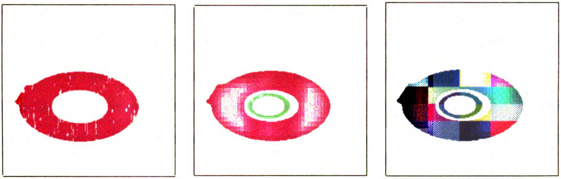
Several issues were explored to detect pits and drops, including the limitations of the sensor, the natural characteristics of defect-free planar surfaces of castings, and the tolerance of the algorithm's implementation to the effects of noise and natural variance of the casting surfaces.

Through tests on real data and Monte Carlo trials on synthetic data, we have attempted to determine, given the expected level of noise in the sensed image, what pit or drop depths can be detected with at least 99% accuracy by our method. We have also determined the probability of a false alarm, i.e., the probability that a cluster of pixels in the image have a variance from the plane which exceeds the planarity tolerance due to random imaging errors.

8.3.1 Segmentation

The segmentation method described in Chapter 5 was used to obtain an initial segmentation of the image. We then subdivided the resulting planar regions. The subdivision allowed us to detect small defects in local regions while still allowing the detection of more "global" defects such as unacceptable tolerances. Initially, we tried to detect defects over the larger surface, but were unable to find most small defects[†] The largest desirable region size was estimated by examining the performance of the surface fitting algorithm over regions of different sizes in real and synthetic data. The present empirical evidence suggests that any region smaller than approximately 2" by 2" is small enough to avoid most of the problems of false rejections and acceptances. However, if the regions are too small, then too much time is spent performing the fitting.

[†] The small defects tended to be obscured by the natural roughness of the casting. This is explained in Section 8.5.1.



(a) Flynn, Jain, and Hoffman segmentation.

(b) New Segmentation (from Chapter 5).

(c) Planar Regions Subdivided Into Smaller Regions.

Figure 8.7. Segmentations of defective gear blank casting.

In the experiments on real data, we subdivided the planar surfaces into regions which are approximately 1.25” by 1.25” in size. Two repartitionings were generated of each image. Each repartitioning was formed by imposing a grid over the planar regions. The grid lines marked the region boundaries of the subdivided regions. We used a spacing of 1.25” between grid lines. The grid lines in the second repartitioning are offset approximately 0.25” from the grid lines used in the first repartitioning. In both repartitionings, any small sub-region less than 0.25” by 0.25” was merged with an adjacent sub-region to prevent unstable fitting in small regions (especially in small regions that contained defects).

The Flynn, Hoffman, and Jain segmentation of a defective gear blank is shown in Figure 8.7(a). The extended method's new segmentation of that same image is shown in Figure 8.7(b). A repartitioning of the large planar surface in that image is shown in Figure 8.7(c).

8.3.2 Plane Fitting

Planes were initially fit using a principal components analysis of local areas of sensed range images. Pixels that deviated from the plane by more than the planarity tolerance were then located. Next, the poor-fitting points were discarded and a new plane was fit to the remaining data. This process was performed for 5 iterations. In most cases, little or no change was observed in the fittings after the second or third iteration, however. After the final iteration, the equation of the plane was returned along with a list of all subregions consisting of multiple connected pixels which deviate from the plane more than the defect threshold.

8.4 Trials on Synthetic Data

8.4.1 Defect-Free Planar Data

Small square planar patches of sizes 0.75", 1.0", 1.25", 1.5", 2", and 3" were synthesized and subjected to additive random Gaussian noise with different noise levels (variances) and zero mean [‡] The planar patches were oriented at several angles with respect to the $x - y$ plane, including inclinations of 0°, 15°, 30°, and 45° (rotations about the y axis). Noise levels of $\sigma = 0.003, 0.004, 0.005, 0.006, 0.007, 0.008$, and 0.009 were used. In the real planar data of the castings acquired using the White scanner, noise levels of σ between 0.003 and 0.005 were observed for all patches smaller than 2" square.

200 Monte Carlo trials were conducted for each possible combination of the plane size, noise level, and orientation. In all cases for $\sigma < 0.007$, the technique for plane fitting and defect detection found no defects on the planar patches in any of the trials. For cases with noise $\sigma \geq 0.007$, Table 8.4 summarizes the results. High levels of noise are clearly a complicating factor in defect detection, but for the lower levels of noise ($\sigma < 0.007$) that are typical of all of the real data collected, noise does not seem to cause false alarms.

[‡] A test for normality of the additive random Gaussian noise was performed to ensure that the noise was truly normal. The density of the distribution and the normal Q-Q plot both appeared to be nearly normal. A Chi squared Goodness-of-Fit test with 19 degrees of freedom (20 classes) was also performed and the hypothesis of normality could not be rejected at the .05 (or lower) significance level(s). (The Chi-squared value was 30.03 and the cutoff was 30.1, however.)

Noise	Orient.	Patch Size					
		.75”	1”	1.25”	1.5”	2”	3”
$\sigma = 0.007$	0°	0%	0%	0%	0%	0%	1%
	15°	0%	0%	0%	0%	0.5%	0%
	30°	0%	0%	0%	0%	0%	0%
	45°	0%	0%	0%	0%	0%	0%
$\sigma = 0.008$	0°	0%	1%	0.5%	1.5%	1.5%	7%
	15°	0%	0%	1.5%	1%	0%	3.5%
	30°	0%	0%	0%	0%	0%	0%
	45°	0%	0%	0%	0%	0%	0%
$\sigma = 0.009$	0°	5%	7.5%	10.5%	14.5%	26%	45%
	15°	3%	4%	5%	7%	14%	28%
	30°	0.5%	0.5%	0.5%	0.5%	1.5%	4.5%
	45°	0%	0%	0%	0%	0%	0%

Table 8.4. Rates of false detection of defects in defect-free synthetic planar data.

This indicates that it is extremely unlikely for random noise from the imaging process and natural grain to cause false alarms in defect-free castings.

8.4.2 Planar Data with One Defect

200 Monte Carlo trials were also conducted on data containing defects (a single pit). Planar data with a single defect were generated for each of the combinations of size, noise level, and orientation that were tested in section 3.1 on defect-free planar data. The single defect was created in the central region of the synthetic plane. It was 0.035” deep, 0.052” in length and 0.045” in width and contained two pixels [§] The six 4-neighbors of those pixels were also made to be depressed by 0.01” from the other pixels in the plane.[¶] Then, random Gaussian noise of mean zero and standard deviation σ was added to all of the pixels in the plane.

The rate of detected defects is indicated in Table 8.5. Ideal performance would be 100%

[§] There are approximately 39 pixels per inch in the x direction and approximately 22 pixels per inch in the y direction in the synthetic images.

[¶] This is typical of pit defects that have been observed in real images.

Noise	Orient.	Patch Size					
		.75"	1"	1.25"	1.5"	2"	3"
$\sigma = 0.003$	0°	100%	100%	100%	100%	99.5%	100%
	15°	99%	99%	99.5%	100%	100%	100%
	30°	94%	94.5%	96.5%	95%	98%	97%
$\sigma = 0.004$	0°	97%	98.5%	99%	99.5%	99.5%	99%
	15°	97.5%	98.5%	98.5%	97.5%	97.5%	97.5%
	30°	88%	93%	85%	88%	86%	86.5%
$\sigma = 0.005$	0°	93%	94.5%	97%	95%	98%	95%
	15°	95.5%	94.5%	93.5%	91%	92%	94.5%
	30°	80%	78.5%	73.5%	80.5%	78.5%	79.5%
$\sigma = 0.006$	0°	87%	90%	92%	92.5%	90.5%	90.5%
	15°	85.5%	89%	87%	82.5%	90.5%	87%
	30°	73%	72%	75%	72.5%	74.5%	75.5%
$\sigma = 0.007$	0°	85%	87.5%	87.5%	83%	85.5%	85%
	15°	81%	84%	80.5%	84.5%	82%	82.5%
	30°	65%	61%	59%	60%	70%	65%
$\sigma = 0.008$	0°	78%	83.5%	82.5%	80%	85%	83%
	15°	82%	75.5%	82%	79.5%	77.5%	80.5%
	30°	60.5%	59%	56%	57.5%	59.5%	61.5%
$\sigma = 0.009$	0°	78%	81%	80.5%	80.5%	83.5%	90%
	15°	75%	78.5%	81%	78%	82.5%	87%
	30°	63%	56.5%	62%	64%	55%	56%

Table 8.5. Rates of correct detection of defects in synthetic planar data with one defect of .035" depth.

detection of the defect. For low levels of noise, very few of the defective samples are accepted. Greater inclination angles tend to decrease the likelihood of detecting the defect.

A similar set of experiments was conducted on synthetic data that contained pits of 0.04" depth. The results are summarized in Table 8.6. The deeper pits are detected very reliably, except in cases of high noise levels coupled with surfaces that were oriented non-horizontally.

Noise	Orient.	Patch Size					
		.75”	1”	1.25”	1.5”	2”	3”
$\sigma = 0.003$	0°	100%	100%	100%	100%	100%	100%
	15°	100%	100%	100%	100%	100%	100%
	30°	100%	100%	100%	100%	100%	100%
$\sigma = 0.004$	0°	100%	100%	100%	100%	100%	100%
	15°	100%	100%	100%	100%	100%	100%
	30°	100%	99%	99%	99.5%	99.5%	100%
$\sigma = 0.005$	0°	100%	99%	99.5%	100%	100%	100%
	15°	99.5%	100%	99.5%	99.5%	98.5%	99.5%
	30°	97.5%	98%	100%	99.5%	99.5%	98.5%
$\sigma = 0.006$	0°	98.5%	99.5%	100%	100%	99%	99%
	15°	97%	98%	99.5%	99.5%	98.5%	99%
	30°	94%	90.5%	94.5%	94.5%	93.5%	95%
$\sigma = 0.007$	0°	97%	98%	96%	99%	95.5%	98%
	15°	96%	98%	97.5%	96.5%	97%	95.5%
	30°	82.5%	91.5%	86.5%	89.5%	87.5%	88.5%
$\sigma = 0.008$	0°	89.5%	93.5%	96%	93.5%	96.5%	93%
	15°	93%	95.5%	92%	94.5%	93.5%	96.5%
	30°	86%	86.5%	87%	83%	84%	87%
$\sigma = 0.009$	0°	93%	91%	95%	92.5%	93.5%	98%
	15°	89.5%	88%	93.5%	94%	93%	94%
	30°	82%	82%	81.5%	80%	79.5%	80%

Table 8.6. Rates of correct detection of defects in synthetic planar data with one defect of depth 0.04”.

Figure	Id.	Pixels	Size	μ	σ	Min.	Max.
8.8(a)	2-a	884	0.8" × 1.2"	2.7×10^{-7}	0.00465	-0.0207	0.0125
8.8(b)	2-b	1266	1.35" × 1"	4.0×10^{-6}	0.0058	-0.0198	0.1910
8.8(c)	2-c	1986	1.8" × 0.8"	-7.7×10^{-6}	0.0032	-0.022	0.011

Table 8.7. Planar patch statistics of defect-free surface.

8.5 Trials on Real Data

To test the validity of the approach, we initially tested data drawn from 3 different castings, one of which was defect-free while the others contained small pits. The estimated depths of the two pits were 0.035" and 0.060".[‖] When the planes were fit to small regions (less than about 1.5" by 1.5"), no defect-free regions are flagged as defective and no defective regions are accepted as defect-free.

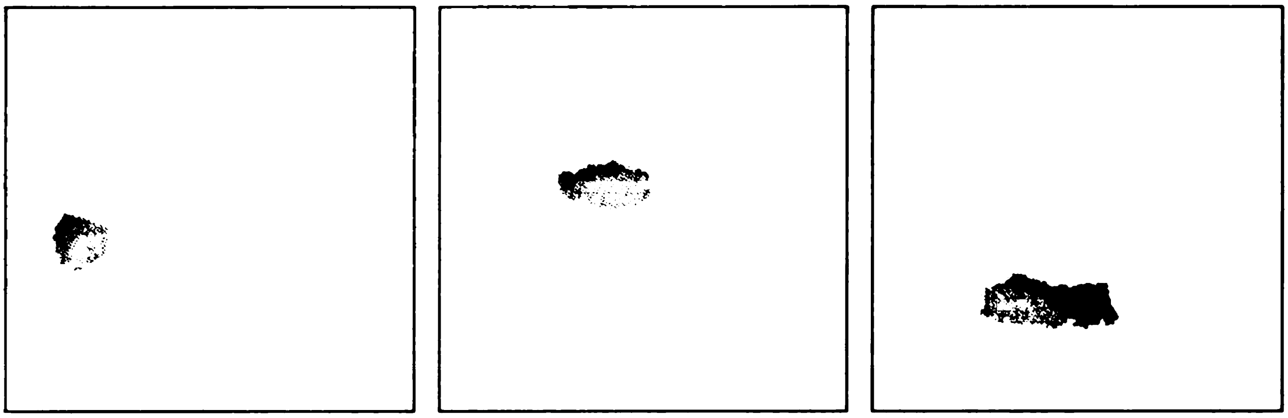
Three defect-free planar patches are displayed in Figures 8.8(a), (b), and (c). All these patches are extracted from different regions of the same planar surface of a gear blank 356 casting. The planar surface is oriented approximately parallel to the $x - y$ plane. Another defect-free region of a different casting, referred to as Patch 1, was shown in Figure 8.2.

Several statistics about the patches in these Figures are indicated in Table 8.7. A plane was fit to each patch using the method of principal components.[**] The Mean Deviation (μ) was the average deviation of the depth values of the points from the plane. Standard Deviation σ, and minimum and maximum deviations (Min. and Max., respectively) were computed on these deviations.

The equations of the planes fit to these sets of data are shown in Table 8.8. Only the plane equations from the first and fifth iterations of the fitting algorithm are shown in the table. The table also indicates the number of points used in the computation of the

[‖] We did not have a method to measure these pits other than the range information collected from the White Scanner. The deeper of the two pits was a "void" region in all range images, forcing us to estimate its depth (it is at least half again as deep as the other pit, but probably not quite twice as deep).

[**] PCA was introduced in Chapter 5.



(a) Defect-free planar patch (Region: 2-a).

(b) Defect-free planar patch (Region: 2-b).

(c) Defect-free planar patch (Region: 2-c).

Figure 8.8. Defect-free planar patches from a range image of a gear blank casting.

Region Identifier	Iteration	Pixels	Plane Equation			
			a	b	c	d
2-a	1	884	-.0203	.0121	.9997	-2.264
	5	848	-.0183	.0109	.9998	-2.271
2-b	1	1266	-.0110	.0270	.9996	-2.221
	5	1153	-.0094	.0230	.9997	-2.239
2-c	1	1986	.00417	.00731	.99997	-2.329
	5	1976	.00418	.00737	.99996	-2.328

Table 8.8. Equations of planes fit to Regions 2-a, 2-b, and 2-c of a defect-free casting. Plane equation is $ax + by + cz = d$.

plane. Although the differences in the equations seem very small, they are actually quite significant. All of the planes are more or less parallel to the $x - y$ plane, but there is more than 0.03" difference in the height of the planes, which is in excess of our algorithm's tolerance for detecting a defect. This underscores the necessity of subdividing large planar surfaces into smaller regions which are tested individually. Our algorithm did not detect defects in any of the three regions.

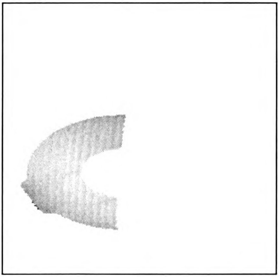


Figure 8.9. Range image of half of a defect-free casting (Region 1-d).

8.5.1 A Large Defect-Free Region

The importance of fitting planes to small regions of the casting is emphasized by our experiments. If a plane is fit over a large region of the casting, local "dips" and "crests" in the surface cause occasional false detection of errors according to our algorithm. Planes fit to smaller sub-regions of the casting are more robust for the purpose of defect detection.

Figure 8.9 is an example of part of a defect-free casting (Region 1-d) that illustrates this phenomenon. If a plane is fit to this entire region, the standard deviation of the points from the plane is $\sigma = 0.0097$ and several "defects" are identified on the casting. Figure 8.10 shows the location of these detected defects over a wireframe of the casting outline.

If the planar surface of this casting is segmented and then subdivided into small regions, no defects are found, however. The subdivision of the casting (Region 1-d) is shown in Figure 8.11. When a plane is fit separately to each of the regions, there is less deviation of the pixels from the local planes. The largest standard deviation of points from a plane in any region was 0.0049 for this image.

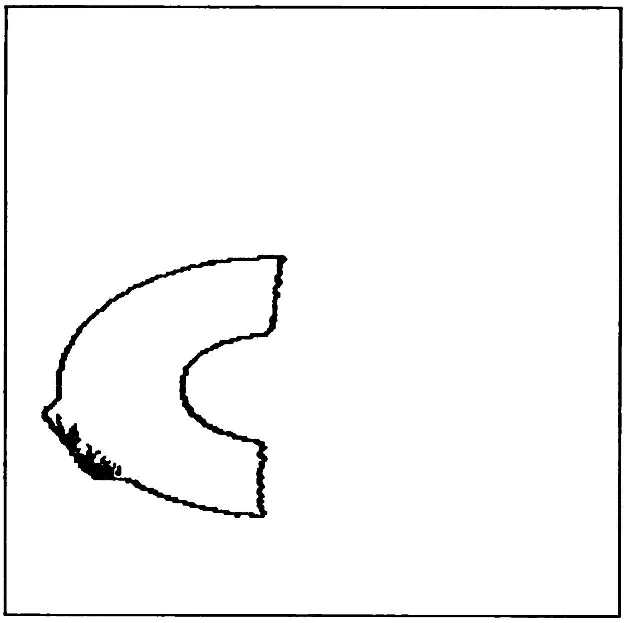


Figure 8.10. Location of defects on casting (Region 1-d), in Figure 8.9.

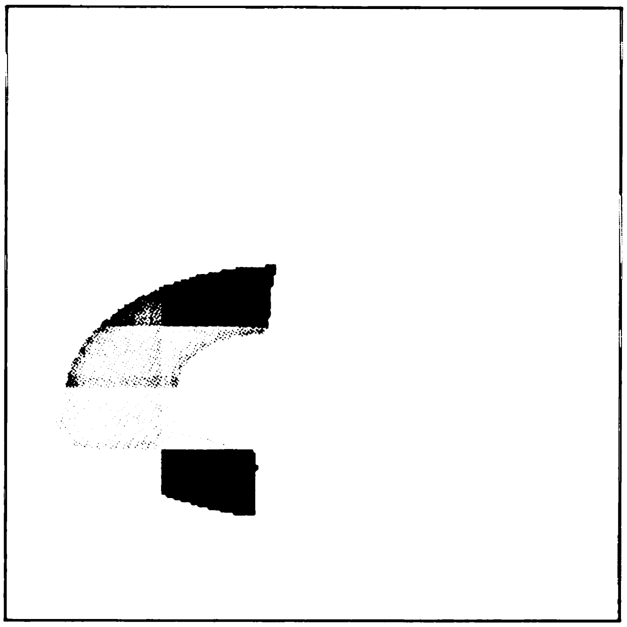


Figure 8.11. Repartitioning of casting (Region 1-d), in Figure 8.9.

Identifier	Defects Detected	Inspection Errors
Big Pit	0 / 4	4 / 4
2 Depress	8 / 8	0 / 8
Blow-Hole	6 / 6	0 / 6
Blow-Hole 2	3 / 6[a]	3 / 6
Pits & Excess	0 / 5	5 / 5
Poured Short	5 / 5	0 / 5
Small Pits	0 / 4	4 / 4
Good A	1 / 4	1 / 4
Good B	0 / 3	1 / 3

Table 8.9. Surface shape inspection for local deviations from planarity. Number of images for which local deviation from planarity defect detected versus total number of images tested is displayed for each gear blank.

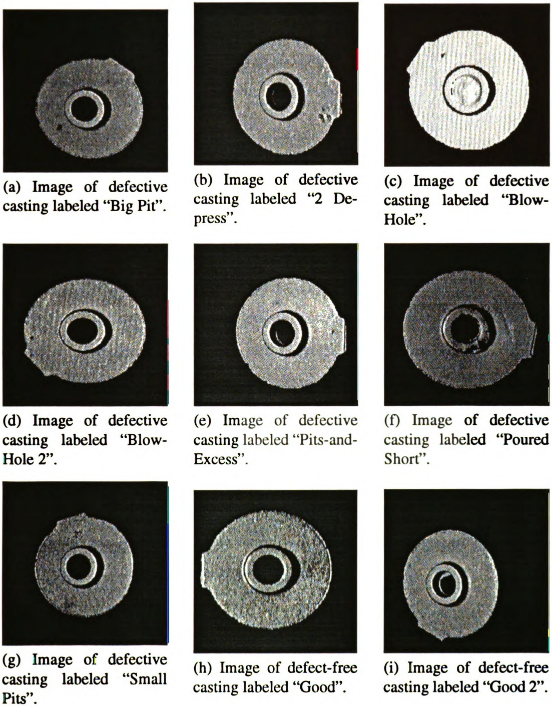
[a] The methods of Ch. 8 were able to detect defects in 2 more cases, ultimately allowing the detection of the defect in all but 1 image.

8.5.2 Planar Defect Detection on Gear Blank 356 Castings

We applied this technique to 45 images of 9 gear blank 356 castings. One image of each of these castings is shown in Figure 8.12. Each of the 9 castings is assigned an identifying label and these are also shown in the image. The inspection results are shown in Table 8.9. Although the results may appear poor, most of the pits in the images appear as "voids" and are detected by the method presented in Chapter 9. Ultimately, the defects were detected in all but one image of the "Blow-Hole 2" casting. One of the images of a defect-free casting ("Good A") was mistakenly found to contain a defect, however. (Only two of the three images of "Good B" do not contain defects. In the third image, a small amount of excess material is visible in the through-hole.) The method of Chapter 9 also mistakenly found a defect in another of the defect-free casting images, though.

Thus, the defects in only 1 of 39 images of defective castings were misclassified as defect-free and 2 of the 6 images of defect-free castings were considered to contain defects.

We also inspected for the planarity of each of the local regions. The mean absolute deviation from planarity for the points in each local region was computed. If that devi-



(a) Image of defective casting labeled “Big Pit”.

(b) Image of defective casting labeled “2 Depress”.

(c) Image of defective casting labeled “Blow-Hole”.

(d) Image of defective casting labeled “Blow-Hole 2”.

(e) Image of defective casting labeled “Pits-and-Excess”.

(f) Image of defective casting labeled “Poured Short”.

(g) Image of defective casting labeled “Small Pits”.

(h) Image of defect-free casting labeled “Good”.

(i) Image of defect-free casting labeled “Good 2”.

Figure 8.12. Images of gear blank 356 castings.

Identifier	Planarity OK / No. Images
Big Pit	4 / 4
2 Depress	7 / 8
Blow-Hole	0 / 6
Blow-Hole 2	6 / 6
Pits & Excess	5 / 5
Poured Short	0 / 5
Small Pits	4 / 4
Good A	4 / 4
Good B	3 / 3

Table 8.10. Surface shape inspection for planarity of sub-regions. Number of images for which no planarity defect is detected for any sub-region versus number of images tested.

ation exceeded an allowable measure, the local region was considered to be non-planar. Specifically, we tested for this using a tolerance of 0.0075" in all of the repartitioned regions.

The results of this test on the 45 images of castings is shown in Table 8.10. The test is not very powerful, although it was able to detect the non-planarity in the center of the "Blow-Hole" casting. (That casting has a very rough region in its center where poured metal filled what should have been a through-hole.) It was also able to detect non-planar regions on all of the "Poured Short" casting images in the regions where the metal "dips" from the ideal plane. A shallower depression was also noticed in one of the images of the casting labeled "2 Depress." The major reason this test does not appear to be very successful in many of these images is that most of the pits are too small to influence the mean deviation of the region that contained them.

8.6 Theoretical Expectation of Performance

If the deviations from true planarity of a planar surface in a casting range image follow a Gaussian distribution, then it is rather straightforward to compute the probability of falsely detecting a defect in a defect-free region. The real defect-free planar casting range data collected with the White Scanner exhibited standard deviations in the range

Region Identifier	σ	Approx. Size
1-a	0.0049	0.5" × 3.5"
2-a	0.0046	1.2" × 0.8"
2-b	0.0058	1.35" × 1.0"
2-c	0.0032	1.8" × 0.8"
3-a	0.0045	2" × 1"
6-a	0.0033	0.8" × 0.9"
6-b	0.0038	1.7" × 1.2"

Table 8.11. Standard deviations from planarity in small planar regions.

$0.003 < \sigma < 0.006$ over small planar regions (of approximate size 1.5" by 1.5"). Table 8.11 shows the standard deviations and region sizes of seven small defect-free planar regions extracted from range images of the three gear blank 356 castings used in this experiment. Images of three of these regions (2-a, 2-b, and 2-c) were shown in Figure 8.8. Larger deviations were only observed when the planar regions were larger than 2" × 2" or when defects were present. For instance, in a doughnut-shaped planar region of approximately 2.3" radius with a center hole of approximate radius 1" (e.g., the large planar surface of the top of the gear blank 356 casting), a standard deviation of 0.005 to 0.008 was commonly observed.

The planar fitting technique is designed to test for defects in small regions, approximately 1" by 1" in size. If two or more adjacent (connected) points that deviate more than 0.025" from the fitted plane are detected in the image, they are flagged as pit defects. If a Gaussian noise model is assumed with zero mean and a reasonable σ standard deviation, there is very little probability that even a single pixel in a small region will deviate more than 0.025" from the plane. For instance, if it is assumed that $\sigma = 0.003$, this probability is equal to the area in the tails of a standard normal curve farther than 8.333 standard deviations from the mean. This area is almost nil. Even if σ is assumed to be as large as 0.005, this probability is still almost negligible (5.7×10^{-7}).

σ	$P(d(P, Z) > 0.025)$
0.004	4.09×10^{-10}
0.005	5.733×10^{-7}
0.006	3.09×10^{-5}
0.007	0.000355
0.008	0.001778
0.009	0.005469

Table 8.12. Probability of pixel deviation for various levels of σ.

The probability that one point deviates more than 0.025" from the plane is

$$P(|\text{deviation}| > 0.025) = 2 * P(\text{deviation} > 0.025) = 2\left[1 - \int_{-\infty}^{0.025/\sigma} \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\, du\right].$$

Table 8.12 contains the probabilities that a pixel P deviates more than 0.025" from a plane Z for various values of σ.

The probability that a defect would be falsely detected by our algorithm is approximately equal to the probability that two (or more) adjacent pixels both (all) deviate more than 0.025" from the plane. For most of the images collected, 1" by 1" square planar regions contain 39 pixels in the x direction and 22 pixels in the y direction. The probability of falsely detecting a defect within this region is, therefore, the probability that at least two adjacent pixels within the region both deviate more than 0.025" from the plane.

This probability is

$$\begin{aligned} P(\text{false alarm}) &= 1 - [1 - 4 * P(0.025, \sigma)]^{37*21} * [1 - 3 * P(0.025, \sigma)]^{21} * \\ &\quad [1 - 2 * P(0.025, \sigma)]^{21} * [1 - P(0.025, \sigma)]^{38}, \end{aligned}$$

where $P(0.025, \sigma)$ is the probability, given a noise level σ, that a pixel deviates more than 0.025" from the plane. The first term of this expression is the probability that no adjacent pixels vary more than 0.025" in the subregion formed by removing the first and last columns

σ	$P(\text{false alarm})$
0.004	5.2×10^{-16}
0.005	1.067×10^{-9}
0.006	3.1×10^{-6}
0.007	0.000410
0.008	0.0102
0.009	0.093

Table 8.13. Probability of falsely detecting defect(s) in a 1" square planar region.

and the top row. The second term is the probability that there are not two adjacent pixels deviating more than 0.025" in the last (rightmost) column. The third term is the probability that there are not two adjacent pixels deviating more than 0.025" in the first (leftmost) column. The last term is the probability that two defective adjacent pixels do not appear in the top (uppermost) row.

Table 8.13 contains the computed probabilities that the defect-free 1" by 1" area contains a defect (due to the addition of random noise caused by the natural roughness of the casting and the imaging process). For small regions, the probability of falsely detecting a defect in a planar region of a casting range image is negligible. At the higher noise levels which are observed in larger regions of the casting, the technique proposed here would probably cause too many false alarms. This is another reason that it was necessary to fit planes to small regions.

8.7 Detection of Defects on Cylindrical Surfaces

This section discusses our method for detecting cylindricity defects and also pit defects on cylindrical surfaces. The method assumes that cylindrical surfaces have been segmented and classified using the technique of Chapter 5. First, the approach is described and then its performance on several real range images is presented.

Initially, the general technique for detection of defects on planar surfaces was applied to cylindrical surfaces as well. Specifically, the cylindrical regions supplied by the output of the segmentation scheme were subdivided into approximately 1.25" × 1.25" regions and cylinders were separately iteratively fit to each region. This technique produced poor results, however, since the regions created by the subdivision were too small to support reliable fits. The model-driven fitting was poor and Flynn's data-driven fitting was even worse.

Thus, we decided not to subdivide the cylindrical regions. The iterative fitting was instead applied to the cylindrical segments. Five iterations were performed, but in practice 2 or 3 iterations were almost always sufficient. The region's parameters were selected from the iteration that exhibited the smallest error. The original segment was then constricted by one pixel and searched for deviations from cylindricity. The region was constricted because pixels near limb edges tend to have noisy depth values and produce too many false alarms. Any set of two or more connected pixels that deviated more than the tolerances for excess or insufficient material were considered to be cylindricity defects.

This technique is probably sufficient to locate large cylindricity defects caused by short pours or drops, but is not very useful in locating pit defects. If a threshold of 0.025" is used to signal pit defects, few of the pits are located and many false alarms (usually near limb edges) are detected.

Instead of using a global threshold, we used the geometric definition of a pit to detect blow holes. A pit can be defined as a very small region of the part that deviates approximately 0.025" or more from the surrounding pixels. When the complete region was searched for pixels that deviated more than 0.025" from the fitted cylinder, the pits were usually "lost" in the natural roughness of the surface; a pit might be 0.025" deeper than the region around it but still deviate only 0.015" from the fitted cylinder. So, the collection of deviations in fit (i.e., the residuals for each pixel) was examined using a "difference filter" that, for each pixel p_i of the region, returned the maximum difference in deviation between p_i and the points in a neighborhood N_i about p_i. We applied the difference filter on the cylindrical

Identifier	Cylindricity (Pit) Defect / No. Images
DR	2 / 2
Big Cyl	4 / 6
Def-Free	0 / 2
Cyl	0 / 3

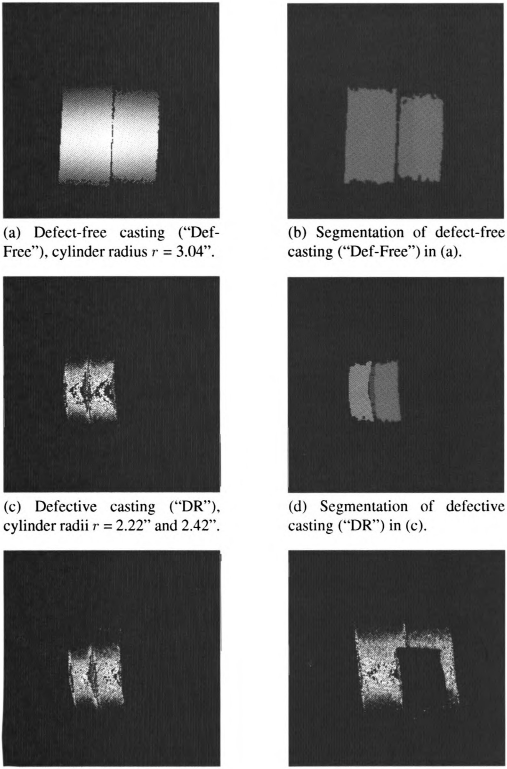
Table 8.14. Cylindricity inspection using 5 × 5 difference filter. Number of images for which cylindricity defect detected versus number of images tested.

regions after constricting them twice (to avoid the problems of noisy depth measurements near the limb edges).

Two sets of experiments were conducted to verify the suitability of this approach. In the first set, we used a difference filter of size 5 × 5. If a pixel p_i deviated -0.025" from any other pixel in the 5 × 5 neighborhood N_i, p_i was classified as a defect. This approach was tested on 8 different images of 2 castings with pit defects in cylindrical surfaces and on 5 images of 2 defect-free cylindrical castings. An image of each of the 4 castings, with segmentations of 2 of them, is shown in Figure 8.13. The radii of these cylinders are shown in the caption of each image. Figure 8.14 shows the collection of deviations from the fitted cylinder for the image in Figure 8.13(c). The rates of error detection are reported in Table 8.14. No defects were detected in any of the defect-free images ("Def-Free" and "Cyl"). Six of the 8 defects were detected.

In the second set of experiments, a difference filter of size 3 × 3 was used with a deviation threshold of -0.02". The results on the same set of 13 images is shown in Table 8.15. No defects were detected in any of the defect-free images ("Def-Free" and "Cyl"). Out of 8 defects, 7 were detected. A defect would have been detected in one of the "Cyl" images if the threshold was lowered to -0.018", however. (The "Cyl" images contain a small pit whose depth is less than the 0.025" threshold for a pit defect.)

The 3 × 3 filter seems to yield slightly better results, although the test set size was quite small. Unfortunately, pit defects on cylindrical surfaces are not very common and only a



(a) Defect-free casting ("Def-Free"), cylinder radius r = 3.04".

(b) Segmentation of defect-free casting ("Def-Free") in (a).

(c) Defective casting ("DR"), cylinder radii r = 2.22" and 2.42".

(d) Segmentation of defective casting ("DR") in (c).

(e) Defect-free casting ("Cyl"), cylinder radii r = 2.22" and 2.42".

(f) Defective casting ("Big Cyl"), cylinder radius r = 3.04".

Figure 8.13. Images of 2 defective and 2 defect-free cylindrical castings.

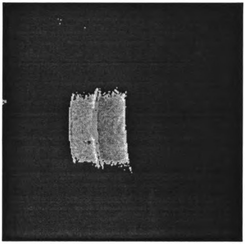


Figure 8.14. Deviations from cylindricity for an image of casting "DR". Brightest points are largest positive deviations (i.e., excess) and darkest points are largest negative deviations (i.e., insufficient material).

Identifier	Cylindricity (Pit) Defect / No. Images
DR	2 / 2
Big Cyl	5 / 6
Def-Free	0 / 2
Cyl	0 / 3

Table 8.15. Cylindricity inspection using 3 × 3 difference filter. Number of images for which cylindricity defect detected versus number of images tested.

small number of defective castings was available.

8.8 Conclusions and Future Directions

In this chapter, we have presented a set of techniques useful for detection of errors on planar and cylindrical surfaces, the two most common surfaces on castings. These techniques are useful to inspect for planarity and cylindricity as well as common casting defects such as insufficient or excess material. Coupled with the techniques of Chapters 7 and 9, we were able to achieve a very high level of precision in accepting and rejecting defect-free and defective parts, respectively.

In the future, we will examine similar techniques for spherical, conical, and free-form surfaces. It should be straightforward to extend our approach of cylindricity defect detection to the spherical and conical defects. But the spherical, conical, and free-form surfaces do not seem to be very common on castings, however, so it may be difficult to collect a set of defective castings for testing. Small filleted or "blended" surfaces between planar and/or cylindrical surfaces are very common in castings, and thus require special attention. We believe that the techniques presented in this chapter are suitable for inspection of most planar and cylindrical surfaces on castings, and provide for detection of the majority of non-crack defects in the casting process. We have been unable to detect cracks in range images. This is an important area for future research. It is possible that cracks may be detected more reliably in intensity images than in range images.

CHAPTER 9

Inspection of Features and Dimensional Tolerances

This chapter discusses our inspection techniques for feature and dimensional inspection. These are considered the highest-level inspection tasks, and perhaps also the most difficult inspection tasks. We first discuss our techniques for feature inspection and then discuss our techniques for dimensional tolerance inspection. Our collection of techniques is not intended to be complete, in fact there are such a great variety of potential features and geometric dimensions that it would be quite difficult to present a complete set of tools. Thus, the techniques in this chapter represent possible methods for inspecting several of the more common features in the domain of castings.

In this chapter, we wish to demonstrate that our system has feature-based and dimensional inspection capabilities. We first present our techniques for feature-based inspection and then present techniques for dimensional inspection. The distinction between feature and dimensional inspection is slightly blurred—there is a great deal of overlap in the two tasks. We view the feature-based inspection as a local technique that functions on a local region of interest on the object while the dimensional inspections are more general techniques that are applicable both in verifying the dimensions of features and in verifying the global properties of the object.

9.1 Feature Inspection

After the inspection for gross defects, a rule-based inspection for the presence or absence of high level features is conducted. At present, our system is limited to inspection of through-holes. We inspect for presence/absence of expected holes, detect undesired holes, and inspect the shape and size of the hole. We wish to distinguish inspection of high-level features such as holes or slots from inspection of the dimensional tolerances of those features and other entities, such as the distance between surfaces or features. Much of the computer-aided design literature either blurs or does not make this distinction, however. Inspection of dimensions, circularity, and planarity are discussed in the second section of this chapter.

In this section, we present our method for inspection of presence/absence of through-holes and for size inspection of this feature. We view these techniques as potentially applicable during in-process inspection to verify that an operation of interest has succeeded or as part of a larger complete module to verify a finished part.

Our technique for through-hole inspection requires that the range image of the object has been segmented by the method of Chapter 5 and that the correspondence between surfaces in the object's image and surfaces in the CAD model has been established by interpretation tree search or fixturing. Through holes are found in the image by first locating all "voids' in the range image*. Regions with no depth value are either background, noise, or concave portions of an object, so they are strong predictors of both through-holes and a common casting defect—pits (or blow holes). We only consider void regions C_j that are completely within the interior of a single segment R_i of the image. Regions C_j that consist of only one pixel are not considered as candidates, however. This reduces the impact of sensor noise on the inspection problem. In general, the void regions represent either blow holes (which are defects) or through-holes (which are features).

*By "void," we mean a connected region of pixels, each of which has no depth value associated with it.

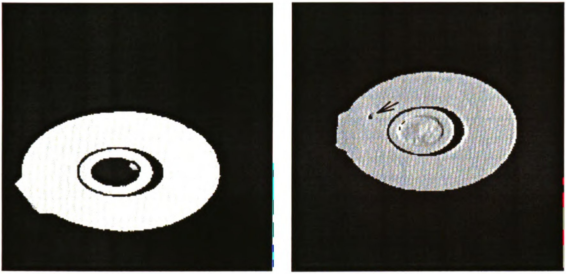
The CAD model contains information about the size and location of the through-holes. This information, as well as the surface(s) that the hole cuts through are extracted from the model. A separate representation that contains acceptable tolerances on these values is queried to generate the inspection plan.

Because the void regions have no depth information, we estimate its parameters by dilating the region C_j by one pixel to form a new region C'_j so that its outline consists of pixels that are members of the surrounding surface region R_i. The center of mass of the outline is used as the estimate of the position of the hole's center. The cross-sectional area (size) of the hole is computed using the parameters of the surrounding region R_i, which we assume to have been estimated using the model-driven method of Chapter 5. At present, we have only computed this area for holes in planar surfaces, although the principle is easily extended to cylindrical, spherical, and conical surfaces. For through-holes in planar regions, each pixel of void region C_j is projected onto the plane $\mathbf{p}$ fit to the surrounding region R_i. The total surface area is estimated as $n_j A$, where n_j is the number of pixels in C_j and A is the area of each rectangular pixel in C_j when it is projected onto plane $\mathbf{p}$. We also determine the radius and position of the circle, although this is presented in Section 9.2, where we discuss dimensional tolerancing.

The inspection module next verifies that the size (surface area) of the circle is within a pre-specified tolerance from the ideal parameters encoded in the model. Any void regions that are outside the limits or which do not correspond to a specified hole in the model are flagged as errors. This means that many pit defects are also detected by this technique. The size of the region is used to determine if the region represents a pit (or a through-hole of insufficient size), a through hole of correct size, or a through-hole of excessive size.

9.1.1 Application of the Technique

A defect-free example of one of the GM castings is shown in the range image in Figure 9.1(a). This image is a top view of a gear or pulley blank, identified as gear blank 356. A defective



(a) Range Image A-1 of Defect-free Gear Blank Casting.

(b) Range Image D-1 of Defective Gear Blank Casting.

Figure 9.1. Range images of two castings.

casting is shown in Figure 9.1(b). This casting has a pit which appears as a void in the range image. The location of the void is marked with an arrow. (The arrow has been overlaid on the casting image and is not actually a part of the casting.)

We tested the technique for hole inspection on a total of 45 images of 9 gear blank 356 castings. One image of each of these castings is shown in Figure 8.12 along with the label we assign to each casting.

Table 9.1 shows the performance of the hole inspection and pit detection for these images. The central through-hole is correctly classified as being acceptable in 34 of 38 cases and correctly classified as being the wrong size (not present) in all 7 images (the 6 images of the casting labeled "Blow-Hole" and one image of the casting labeled "Good B"). "Good B" contains a small amount of excess material within its through-hole although this defect is only visible in one of the three images. Four of the images of the "Poured Short" casting were classified as having a through-hole of incorrect size, but this apparent misclassification is acceptable because the casting is defective anyway. In fact, one of the

Identifier	Through Hole OK / No. Images	Pit(s) Detected / No. Images	Inspection Errors / No. Images
Big Pit	4 / 4	4 / 4	0 / 4
2 Depress	8 / 8	8 / 8	0 / 8
Blow-Hole	0 / 6	6 / 6	0 / 6
Blow-Hole 2	6 / 6	2 / 6[a]	4 / 6
Pits & Excess	5 / 5	5 / 5	0 / 5
Poured Short	1 / 5	4 / 5	1 / 5[b]
Small Pits	4 / 4	4 / 4	0 / 4
Good A	4 / 4	0 / 4	0 / 4
Good B	2 / 3	0 / 3	0 / 3[c]

Table 9.1. Feature inspection results. Number of images satisfying given criteria versus total number of images is displayed for each gear blank.

[a] The methods of Ch. 8 were able to detect defects in 3 more cases, ultimately allowing the detection of the defect in all but 1 image.

[b] The methods of Ch. 8 were able to detect the defect in the one image where it was not detected here.

[c] The defect is visible in only two of the images of Good B. The image with the defect was recognized as a defect.

defects in this casting was in the lip around the through-hole, making it difficult to detect the through-hole, so these errors are acceptable. Pits were detected in 33 out of 38 cases in which they were present [†] and were not detected in the 7 cases where they were not present (in the defect-free images of "Good A" and "Good B"). For the 5 cases where pits were not detected, the methods of Chapter 8 could detect errors in 4 of them.

Thus, the techniques presented here are very useful for inspection of features, specifically, detection of the presence or absence of through-holes. The technique also allowed detection of pits or blow holes in the surfaces.

[†] The "Poured Short" casting actually does not contain pits, but it contains a depression which could be viewed as a large pit.

9.2 Dimensional Tolerance Inspections

When an object is designed using a CAD system, the collection of part dimensions forms the specification of an ideal part. But variances in the production process make it impossible to build a part in exact accordance with the specification. Thus, the design is usually augmented with information defining acceptable limits on size and shape variations. These limits are referred to as tolerances[‡] Tolerances describe the nature and size of "features" based on roundness, cylindricity, position, perpendicularity, etc. [111] (although in this context, "features" need not be strictly equivalent to the features present in a feature-based design[§]). Historically, most mechanical design tolerances have been expressed informally and are mathematically non-rigorous [176]. This makes it very difficult to compute tolerances [176], especially for automated inspection applications. The study of formal tolerance specification and analysis has received some attention, especially in the CAD and manufacturing literature [95, 145, 146, 176]. Most of the current tolerance analysis techniques rely on the designer to formulate the problem and explicitly resolve any ambiguities resulting from groups of related features or tolerances [43]. While the importance of including tolerance analysis capabilities in CAD packages has been generally recognized, very few analysis techniques have been provided [43].

One of the problems with tolerancing information is that there is no robust definition of tolerance [111]. Specifically, tolerances usually involve no precise definition of the features involved and do not formally define the allowable "size." One suggestion to resolve the notion of size involves defining the boundary of the surfaces that a measure must lie within. For instance, scale could be specified through a "size zone," while a "form zone" could

[‡] Tolerance limits must be chosen carefully because they have a great impact on the overall manufacturing cost. Turner and Wozny [176] have noted that it is possible for improper tolerances to make up 50% or more of manufacturing costs.

[§] The term "feature" is overloaded and perhaps slightly confusing to the reader. It generally refers to some area of interest on a part, but can have slightly nuanced meanings depending on the context. There are features in the pattern recognition sense (e.g., the input to the CLUSTER algorithm), features that represent a high-level abstraction or mini-component of a design, "features" that are really geometric dimensions or tolerances, and also "features" that are merely points, lines, planes, or other foci of interest on an object.

specify shape and a "position zone" could specify the bounds on location [111]. Requicha has presented several papers addressing this problem and has proposed that tolerances be specified using a volume that defines the maximum and minimum allowable offsets from faces of the object [145, 146]. Requicha's method does not specify the minimum and maximum allowable amounts of material, however. His specification actually defines a region in a high dimensional space [145, 176]. In addition, Requicha's method varies considerably from historical methods of tolerance specification, and it does not allow a unified description of all possible types of tolerances [176]. Turner and Wozny [176] have proposed that tolerances be interpreted as variables that define a real-valued normal vector space of model variations. A region is defined in the vector space using the limits on the tolerance variables. Every object that is constructed within allowable tolerances can be mapped into a point within the region. Turner and Wozny's method requires careful selection of basis vectors for the space, however.

Deciding what actually needs to be inspected to determine tolerance fulfillment is difficult. Merat et al. [114] have formalized the problem of determining the optimal number of inspection points to determine adherence to tolerance as:

> Given a surface S, let $P = \{P_i; 1 \leq i \leq m\}$ and $Q = \{Q_j; 1 \leq j \leq n\}$ be sets of variables representing points on S. Let C be a set of constraints on the points in P and let D be a set of constraints on the points in Q. We want to find values for all P_i and Q_j such that (1) all constraints in C and D are satisfied, and (2) the set $P \cup Q$ is as small as possible—i.e., as many points of Q are identified with points of P, and vice versa.

Although this formulation is aimed at determining points for a coordinate measuring machine (CMM) to sense, it is also applicable to the related problem in vision-based inspection. Apparently, there are only some heuristics available that attempt to achieve this [114].

For any dimensional inspection, it is necessary that one or more datums[¶] be determined for each model so that all tolerances are consistent and unambiguous. Datums might consist of a specific point, line, plane, or other region on a part [111]. It is probably necessary to introduce redundancy in the datums, i.e., define multiple datums, because (1) objects may contain defects and (2) images of the objects may be from viewpoints in which the datum is invisible. For object recognition and inspection applications, Marshall [111] has outlined several requirements for datums. These include:

1. A datum must be robust with respect to the viewpoint of the sensor.

2. A datum plane should have a large surface area.

3. The datum should be on a region whose texture does not make the recovery of depth data problematic.

4. If no single datum is satisfactory, a system of planes should be used.

For most object models, it is necessary to specify one or more primary datums as well as several secondary datums that capture the relationships between part features or regions.

The inspection of dimensional tolerance is not a trivial undertaking. There are many unresolved problems, especially in the determination of datum points and the "stackup" of tolerances. Datums are often not explicit within the design, making it quite difficult to determine how to measure tolerances. We do not address the problems of datum determination and tolerance stack-up in our work. While these are clearly important issues, they go beyond the scope of this dissertation. We wish to demonstrate that automated visual inspection is able to verify tolerances, provided that the datums and the tolerances are available in the model in an explicit form. We assume that such information is available.

Dimensional tolerances will be inspected by extracting the tolerance information from the CAD model and then examining the object's dimensions for adherence. The level of

[¶] Datums are reference points or features. Tolerances are usually measured with respect to one (or more) datum(s).

tolerances we can check are limited by the spatial resolution of our range scanner, however. The White Scanner quantizes y and z coordinates to 0.001", with a standard deviation of noise in the z coordinate of less than 0.01 inch [55].

9.2.1 Tolerance Analysis

Several methods related to tolerance analysis and generation of tolerance inspection plans have been presented in the literature.

Merat et al. [114] have presented a system that allows automatic generation of inspection plans from certain feature-based CAD models. Their system presents the feature-based computer-aided designer with a menu of choices that allows the designer to describe feature dimensions and tolerances. The selected dimensioning technique is translated into a representation that conforms to the ANSI Standard for Geometric Dimensioning and Tolerancing. The selection of a technique also triggers the generation of an appropriate pre-fabricated inspection plan for use by a coordinate measuring machine.

Dong and Soom [43] have proposed a technique for resolution of tolerance stackup through a tree search method. Their technique extracts tolerance information from IGES entities that contain a CAD drawing's text annotations. A tree structure is constructed from this extracted data which implicitly contains the relationships between all tolerances. Search of the tree allows recovery of related tolerances.

9.2.2 Dimensional Tolerance Inspection of Castings

In the remainder of this section, we present our techniques for verification of several tolerances. We inspect for circularity, circle radii, distances between circular arcs, distance between planes, and parallel planes. We also inspect for cylindricity and planarity using the techniques presented in Chapter 8.

Identifier	Orientation OK / No. Images	Separation OK / No. Images
Big Pit	4 / 4	4 / 4
2 Depress	8 / 8	8 / 8
Blow-Hole	6 / 6	3 / 6
Blow-Hole 2	6 / 6	6 / 6
Pits & Excess	5 / 5	5 / 5
Poured Short	3 / 5	1 / 5
Small Pits	4 / 4	4 / 4
Good A	4 / 4	4 / 4
Good B	4 / 4	4 / 4

Table 9.2. Dimensional inspection for planar separation and orientation. Number of images for which there is no defect in any sub-region versus number of images tested.

Planar Separation and Orientation

We inspect for the distance between parallel planes and for the difference in planar orientation by examining the planar surface parameters that were extracted using the methods of Chapter 5. The surface classifications were carried out upon the segmentations generated using the method of Chapter 5. We assume that the correspondence between model and object surfaces has been established. We also assume that the model database has been queried to determine the ideal values and tolerances for these measures.

In the model of the gear blank 356 casting, tolerances of 5° in difference of planar surface orientation and 0.04" in parallel planar separation were used. We performed the inspection in the 45 images of 9 castings shown in Figure 8.12. The results of the inspection are shown in Table 9.2. Except for the "Poured Short" and "Blow-Hole" castings, the planar separations and orientations were determined to be acceptable, which is the correct result. For the "Poured Short" image, the defect causes the small planar "lip" in the center of the casting to be both non-planar and "short" of its desired height. The excess material in the center of the "Blow-Hole" casting is also very rough and, while it is almost as "high" as the surrounding lip, is slightly "short" of the desired height of the lip. The segmentation scheme did not always isolate these defects in a fashion that made it possible to easily detect the planar orientation defect. However, in the few cases where the planarity defect was not detected, the defective nature of the casting could still be recovered (e.g., by detection of insufficient material defects in the region of the defect). It could be argued that the planar separation and orientation defects in these two castings are minor symptoms of the actual errors that we recover in other stages of defect detection.

Circularity Inspection

In this subsection, we describe our technique for circular dimensional inspections, including circle size, position, and circularity (i.e., excess or insufficient material along a circular boundary). We assume that the image regions in which circularity is to be tested are known through their correspondence with model surfaces. A general segmentation of the part is assumed to have been conducted using the technique of Chapter 5. We also assume that the model database has been queried to determine the ideal values and tolerances for these measures.

For the measurement of circular tolerances, we use the center of the through-hole as the datum point on the gear blank 356 casting. The recovery of this datum is important because the model of the gear blank casting measures several dimensions from the center of the part. We estimate this point by first finding the center of mass of the largest planar region P (i.e., the doughnut-shaped region) in images of the casting. This estimate of the center C_1 of the through-hole is then refined.

We sweep an arc of radius r in P's plane about the initial estimate C_1 and mark all pixels in the interior of the swept circumference. The set of marked pixels is then dilated twice. The largest unmarked connected component outside this marked area is then removed from the region P. (This component should correspond to most of the "gate" on the side of the casting.) The new estimate of the center of the through-hole C_2 is then refined to be the center of mass of the new region P.

It may appear to have been more straightforward to estimate the center of the through-hole simply by determining the position of the through-hole and then finding its center of mass, but this did not provide a stable estimate of the center. The initial approaches we attempted to estimate the center were unsuccessful. First, we tried to find the center of mass of the points that make up the innermost boundary of the upper circular "lip" of the casting. Next, we estimated the through-hole's center as the center of mass of the entire lip region. Both of these methods were unstable, probably because very little data was being used to estimate the position of the center. Finally, we attempted to locate the large central void that represents the through-hole and find the center of mass when these points were projected onto the plane that contained the upper lip. This was also unsuccessful due to depth values being returned from some of the points lying inside the cylinder of the through-hole (i.e., some of the points along the wall of the through-hole were visible, especially those near the top of the through-hole next to the upper "lip," thus reducing the size of the through-hole void).

The maximum allowable size of the through-hole was then extracted from the model. A circle of this radius was inscribed in the plane p_u of the upper "lip" region u about the estimated through-hole center C_2^*, where C_2^* is the through-hole center C_2 projected onto the plane p_u. The circumference of this circle was traversed in region u. If any pixels were missing from the circumference, the casting was considered to have a circularity defect.

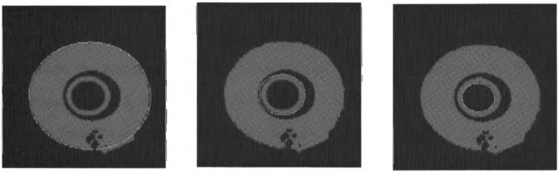
Similar techniques are followed for the other two circles, although since they form the outer boundaries of regions, we first inscribe the minimum size circle to determine if there is enough material present and then inscribe the maximum size circle to make sure that no additional material is present. (If any pixels are missing from the minimum size circle's circumference or if any pixels are present outside the maximum size circle's circumference, a circularity defect is considered to be present.)

In the model of the gear blank 356 casting, we inspected for the presence of three circular outlines—the inner and outer outlines of the innermost central "lip" and the outer outline of the part. We could not inspect the circular boundary of the larger doughnut-shaped

Identifier	Circularity OK / No. Images
Big Pit	4 / 4
2 Depress	7 / 8
Blow-Hole	4 / 6
Blow-Hole 2	6 / 6
Pits & Excess	5 / 5
Poured Short	5 / 5
Small Pits	4 / 4
Good A	3 / 4 [a]
Good B	2 / 3

Table 9.3. Dimensional inspection for circularity. Number of images for which no planarity defect detected for one or more entire sub-region versus number of images tested.

planar region near the lip because occlusion limited the visible portion of this outline. We performed the inspection in the 45 images of 9 castings shown in Figure 8.12. The results of the inspection are shown in Table 9.3. Only the casting labeled "Blow Hole" actually contains a circularity defect—there really is no central circle due to excess material filling the through-hole. An example showing the detected—and acceptable—circular arcs on one of the images of the casting labeled "2 Depress" is shown in Figure 9.2.



(a) Outermost Circular Arc.

(b) Inner Circular Arc.

(c) Innermost Circular Arc.

Figure 9.2. Detected circular arcs overlaid on labeled image of a casting.

[a] An additional defect was detected by the methods of Ch. 8, so only 2 of the 4 "Good A" instances were judged defect-free.

Unlike our testing of the relationship between planes, the circularity dimensional inspection must be carried out on a single region. Sensor limitations complicated our task in this regard. The main problem was the limited spatial resolution of the sensor in the x and y directions. This was probably the chief culprit in the rejection of one instance of each of the defect-free castings ("Good A" and "Good B") and for the one instance of the "2 Depress" casting. In all three cases, only one pixel was found missing from the circumference of one of the three desired circles. Frequently, the tolerance on a measurement is very close to the spatial resolution of the sensor. More dense range data would be necessary for robust performance of dimensional inspection. It may also be possible to achieve better performance by refining the estimation of the circle center slightly using a hill-climbing technique.

9.3 Conclusions

In this chapter, we have discussed some of the issues involved in tolerance specification and analysis. The study of tolerances is an area of considerable interest in the CAD and manufacturing literature. If the tolerances are not specified in a manner such that the inspection algorithms can easily recover them in a usable, explicit format from the model, then the dimensional inspection task becomes more difficult.

We have presented techniques for inspection of casting dimensional tolerances and for common casting features, namely through-holes. Our through-hole inspection techniques also allow the detection of pit defects in castings. When coupled with the techniques of Chapters 7 and 8, we are able to detect most of the defective castings in our testbed of images. Table 9.4 shows the performance of our system on the 45 gear blank casting images. Table 9.5 shows the performance of our system on the 13 cylindrical casting images. The spatial resolution of the White Scanner presented difficulties for our inspection algorithms, however. A better sensor or multiple range images would be necessary to perform a complete inspection of all dimensions.

	Defect Detected	Defect Not Detected
Defect Present	38	1
No Defect Present	2	4

Table 9.4. System performance for gear blank casting inspection.

	Defect Detected	Defect Not Detected
Defect Present	7	1
No Defect Present	0	5

Table 9.5. System performance for cylindrical casting inspection.

The feature inspection technique in this chapter focuses on a specific area of interest within the part, while the dimensional inspection techniques are applicable to the part as a whole. We view these inspection as the high-level inspection tasks in our system. In a complete inspection system, methods to inspect all potential high-level features would have to be built-in.

CHAPTER 10

Conclusions and Future Directions

10.1 Overview

In this dissertation, we have focused on the inspection of 3D objects whose identity is known. A system for the inspection of castings in range images has been developed and examined. Information from the CAD model of the object has been used throughout the inspection system to allow for more reliable performance. The techniques presented are applicable for inspecting objects with planar, cylindrical, spherical, and conical surfaces. Specific inspection tasks performed include inpsection for dimensional tolerances, gross assembly integrity, and common casting defects such as pits, excess material, short pours, etc. The performance of the system has been examined through theoretical models, Monte Carlo trials on synthetic range data, and on many real range images of industrial castings. We believe that this work is one of the first "general-purpose" inspection methods to have been tested on a set of 3D castings using real range images.

We discussed the behavior of our range sensor and the representation requirements for the object models. Segmentation and model-based quadric surface classification techniques were also presented. The model-based surface classification scheme exhibits very good performance, as measured by comparison to a data-driven classification procedure, and has been extensively tested on noisy synthetic and real range images. The surface classification techniques exhibit fitting errors in real data that tend to be very close to the resolution limits of the range sensor. Several possible techniques for pose estimation were also examined, although the current implementations of those techniques may not provide sufficient accuracy for inspection purposes. We also studied the feasibility of parallel computing for real-time object localization and recognition by implementing and analyzing the performance of the interpretation tree search on a MIMD machine.

A bidirectional template matching scheme for detection of gross defects on castings was presented. This scheme is useful for quick in-process inspection and as a verification procedure to check pose hypotheses. Several techniques for detection of excess or insufficient material casting defects were also developed and extensively tested. Finally, we exhibited inspection techniques for verification of dimensional tolerances and design features. Our system contains techniques to inspect for part dimensioning, surface shape defects, and assembly integrity (gross defects). Ultimately, we were able to correctly identify 38 out of 39 instances of defects on planar surfaces in castings and 7 out of 8 instances of defects on cylindrical surfaces in castings. In defect-free castings, the system falsely reported errors on 2 out of 6 planar surfaces and on 0 of 5 defect-free cylindrical surfaces.

10.2 Future Research Directions

There are several open research issues requiring attention in every stage of the inspection process. At the first stage, model building, bugs were encountered in the IDEAS CAD package (which was the only package available to us). IDEAS VI is unable to model fillets without introducing artifacts. It can also be difficult to modify complex CAD models in this CAD package. IDEAS VI seems prone to unexpected crashes, too, probably because of bugs in subroutines for fillet representation. In the future, we hope to create models using a new version of IDEAS that corrects these problems. The 100X range sensor also limits the type of inspection tasks we can perform. This sensor is too slow for real-time inspection, although other faster sensors are available. The sensor's spatial resolution is also inadequate for many dimensional measurements. The sensing arrangement causes many self-occlusions to occur as well. Some of these problems can be overcome using multiple range images, although it is not possible with our current sensing arrangement to "turn over" an object, limiting data collection to only one hemisphere of an object. Registration of multiple range images of a given 3D object is not a trivial problem. Our lack of precision fixtures created additional difficulties in verifying the absolute performance of our system because no ground truth data was available. A more fundamental problem we faced was the difficulty in acquiring objects. We were unable to obtain any castings with conical or spherical surfaces. Furthermore, we were unsuccessful in finding multiple instances of castings with mismatch or warp-like defects.

One of the failures in our segmentation algorithm is that pixels adjacent to region boundaries cannot be labeled. The pixels which are close to the region boundaries, especially pixels near or on filleted or blended surfaces, are often mis-labeled. The depth value of many pixels near limb edges also appear to deviate a large amount from their expected values, producing both segmentation and classification errors. The classification procedure cannot estimate surface parameters reliably for any region which is too small. Conical surfaces with a small vertex angle also cannot be fit reliably. Furthermore, we are unable to estimate the parameters for filleted and blended regions. These difficulties in segmentation and classification probably cannot all be resolved by refinement of the existing segmentation and classification procedures. We are investigating model-based segmentation to address most of these issues. Information from the model may be useful to assign a region label to any unlabeled pixels, and may also be used in a second merging and splitting step that re-segments the current labeling. Another approach we are investigating involves using the initial segmentation and classification to generate a coarse estimate of object pose. If several regions can be extracted reliably, the coarse estimate of pose should be reasonably accurate. After this pose transformation is applied to the model, the surface labeling from the model could be imposed on the range image of the object to re-label any existing mislabeled or unlabeled regions. We are also exploring RANSAC-type extensions and parallel computation for the classification schemes. One very challenging area that we wish to address in the future is matching of free-form surfaces. This is a difficult, but important, problem, especially in inspection of many products constructed of plastic or sheet metal.

Perhaps the weakest stage in our system is the object localization stage. We are exploring iterative refinement of pose estimation and stronger verification steps to ensure better pose estimation. Pose estimation has heavy computational requirements, so we would like to continue to investigate parallel algorithms for pose estimation. The trade-off between stronger and weaker constraints in the interpretation tree search also needs to be studied. In addition, we are eager to explore object recognition, especially in the MIMD environment. It is somewhat difficult to develop, debug, and fine-tune parallel programs, however, so those explorations could be lengthy.

Several extensions to the coarse verification stage of inspection are also envisioned. First, the current verification scheme assumes that pose is restricted through the use of fixtures. Less rigid requirements would make the technique more flexible and generally applicable. The use of the Hausdorff matching is a good first step in this area, although additional exploration of that technique, especially in allowing another degree of rotational freedom, is necessary. We are particularly interested in adapting the coarse verification step into a multi-resolution template approach, in which points that do not fit the template well in the first stage, are used as focus or seed points for a finer resolution template matching. We also aim to develop template-matching schemes for verification of specific operations of interest.

The surface shape inspection techniques function quite well, but are restricted to only planar and cylindrical surfaces. We would like to extend these methods to spherical, conical, and other surfaces if it is possible to obtain castings containing such surfaces. One weakness of the surface shape inspection is that very few cracks can be detected using range sensing.

Detection of cracks in castings is quite important and will be investigated in the future, perhaps using intensity images.

Finally, additional capabilities for feature inspection need to be built into the system. Our system concentrated on inspecting *through-holes* because they were a very common casting feature. Additional castings with other features would have to be acquired to test any other feature inspection methods. The issues of datum determination and tolerance stack-up were ignored in our system, although any practical system would have to address those issues. In the future, we hope to investigate tolerance analysis.

10.3 Summary

We have discovered that for even a fairly straightforward inspection algorithm to function well, the techniques must be carefully tailored to consider the capabilities of the sensor, the interaction and feedback between various earlier stages of the inspection system, and the manufacturing process. This factor may limit the development of true general-purpose inspection systems; true general-purpose inspection may only be possible by collecting many specialized defect detection functions together into one system. Careful attention to the commonly occurring defects in the manufacturing process must be built-in to any inspection technique as well. Perhaps the greatest contribution of this thesis to automated visual inspection is that the inspections conducted are designed to detect actual defects in real industrial parts. This stands in contrast to many of the other experimental inspection systems that sometimes function on contrived defects on only one or two objects.

Finally, general-purpose inspection is a difficult undertaking, especially if free-form surfaces must be inspected. The methods presented in this dissertation provide a good basis from which a more general system could be developed, although there are many unresolved issues with free-form surfaces, tolerance specification and analysis, segmentation, and, to a lesser extent, data and model representations and classification. Unfortunately, these apparently separate issues cannot be completely decoupled. Each of them, especially at the later stages of processing, depends on some or all of the other decisions in the inspection system. General-purpose automated visual inspection is an enticing proposition, but much more progress must be made on these unresolved issues before general-purpose inspection can truly become reality.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] American Foundrymen's Society, *Analysis of Casting Defects, Second Edition*, Des Plaines, IL, 1966.

[2] American Foundrymen's Society, *Casting Defects Handbook*, Des Plaines, IL, 1972.

[3] Robert Ayres, Steven Miller, James Just, Keith King, Michael Osheroff, George Berke, Peter Spidaliere, and Tran Ngoc, editors, *Robotics and Flexible Manufacturing Technologies*, Noyes Publications: Park Ridge, NJ, 1985.

[4] Vivek V. Badami, Nelson R. Corby, Nancy H. Irwin, and Kenneth Overton, "Using Range Data for Inspecting Printed Wiring Boards," *Proceedings of the IEEE International Conference on Robotics and Automation*, Philadelphia, 1988, pp. 1466-1468.

[5] Narinder Bains and Frank David, "Machine Vision Inspection of Fluorescent Lamps," *Proceedings of the Conference on Machine Vision Systems Integration in Industry, SPIE Vol. 1386*, Boston, 1990, pp. 232-242.

[6] Dana H. Ballard and Christopher M. Brown, *Computer Vision*, Prentice-Hall: Englewood Cliffs, New Jersey, 1982.

[7] B. G. Batchelor, D. A. Hill, and D. C. Hodgson, editors, *Automated Visual Inspection*, IFS Publications Ltd.: Kempston, Bedford, England, 1985.

[8] Bruce G. Batchelor and D. W. Braggins, "Commercial Vision Systems," in Torras (ed.), *Computer Vision: Theory and Industrial Applications*, Springer-Verlag: New York, 1992, pp. 405-452.

[9] BBN Advanced Computers, *Programming in C with the Uniform System*, Cambridge, MA, 1988.

[10] Paul J. Besl, "The Free-Form Surface Matching Problem," in *Machine Vision for Three-Dimensional Scenes*, edited by Herbert Freeman, Academic Press: New York, 1990, pp. 25-71.

[11] Paul J. Besl and Ramesh C. Jain, "Three-Dimensional Object Recognition," *Computing Surveys*, Vol. 17, No. 1, March 1985, pp. 75-145.

[12] Paul J. Besl and Ramesh C. Jain, "Segmentation through Variable-Order Surface Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, 1988, pp. 167-192.

[13] Bir Bhanu and Lawrence A. Nuttall, "Recognition of 3-D Objects In Range Images Using a Butterfly Multiprocessor," *Pattern Recognition*, Vol. 22, No. 1, 1989, pp. 49-64.

[14] Dorothea Blostein and Narendra Ahuja, "Shape from Texture: Integrating Texture-Element Extraction and Surface Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 12, 1989, pp. 1233-1251.

[15] Ruud M. Bolle and Daniel Sabbah, "Differential Geometry Applied to Least-Square Error Surface Approximations," *Proceedings of the SPIE Conference on Optical and Digital Pattern Recognition Vol. 754*, Los Angeles, 1987, pp. 117-127.

[16] Robert C. Bolles and R. A. Cain, "Recognizing and Locating Partially Visible Objects, the Local-Feature-Focus Method," *International Journal of Robotics Research*, Vol. 1, No. 3, 1982, pp. 57-82.

[17] Robert C. Bolles and P. Horaud, "3DPO: A Three-Dimensional Part Orientation System," *International Journal of Robotics Research*, Vol. 5, No. 3, Fall 1986, pp. 3-26.

[18] Paul J. Boone, "ROSA III, a Third Generation Steam Generator Service Robot," *Proceedings of the International Robots and Vision Automation Conference*, Detroit, October 1991, pp. 16-59 to 16-70.

[19] Kevin Bowyer, David Eggert, John Stewman, and Louise Stark, "Developing the Aspect Graph Representation for use in Image Understanding," *Proceedings of the DARPA Image Understanding Workshop*, Palo Alto, CA, May 1989, pp. 831-849.

[20] Don W. Braggins, "The Development of the Machine Vision Industry in Europe During the Decade of the 80s," *Proceedings of SME VISION '90*, Detroit, November 1990, pp. 13-19 to 13-27.

[21] "CADEYES 3-D Measuring System," product brochure, Air Gage Company, Livonia, MI, 1993.

[22] Andrea Califano, Rick Kjeldsen, and Ruud M. Bolle, "Data and Model Driven Foveation," *Proceedings of the Tenth International Conference on Pattern Recognition*, Atlantic City, June 1990, pp. 1-7.

[23] X. Cao, N. Shrikhande, and G. Hu, "A Quadric Data Fitting Method Using Orthogonal Distance Measure," Central Michigan University Computer Science Department Technical Report.

[24] L. Caponetti, A. Distante, N. Ancona, and R. Mugnuolo, "Three-dimensional Object Recognition based on Multiview Representation and Extended Gaussian Image," in *Real-Time Object Measurement and Classification*, ed. by Anil K. Jain, Springer-Verlag: New York, 1988, pp. 227-249.

[25] J. P. Chan, B. G. Batchelor, I. P. Harris, and S. J. Perry, "Intelligent Visual Inspection of Food Products," *Proceedings of the SPIE Conference on Machine Vision Systems Integration in Industry Vol. 1386*, Boston, November 1990, pp. 171-179.

[26] C. Chen and A. Kak, "A Robot Vision System for Recognizing 3-D Objects in Low-Order Polynomial Time," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 6, 1988, pp. 1535-1563.

[27] Homer H. Chen and Thomas S. Huang, "Maximal Matching of 3-D Points for Multiple-Object Motion Estimation," *Pattern Recognition*, Vol. 21, No. 2, 1988, pp. 75-90.

[28] Jin-Long Chen, George C. Stockman, and Kashi Rao, "Recovering and Tracking Pose of Curved 3D Objects from 2D Images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, June 1993, pp. 233-239.

[29] Sei-Wang Chen and Anil K. Jain, "Object Extraction from Laser Radar Imagery," *Pattern Recognition*, Vol. 24, 1991, pp. 587-600.

[30] Shuang Chen and Herbert Freeman, "On the Characteristic Views of Quadric-Surfaced Solids," *Proceedings of the IEEE Workshop on Directions in Automated CAD-Based Vision*, Maui, Hawaii, June 1991, pp. 34-43.

[31] Roland T. Chin, "Automated Visual Inspection: 1981 to 1987," *Computer Vision, Graphics, and Image Processing*, Vol. 41, 1988, pp. 346-381.

[32] Roland T. Chin, "Automated Visual Inspection Algorithms," in Torras (ed.), *Computer Vision: Theory and Industrial Applications*, Springer-Verlag: New York, 1992, pp. 377-404.

[33] Roland T. Chin and Charles R. Dyer, "Model-Based Recognition in Robot Vision," *Computing Surveys*, Vol. 18, No. 1, March 1986, pp. 67-108.

[34] Roland T. Chin and Charles A. Harlow, "Automated Visual Inspection: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, 1982, pp. 557-573.

[35] Ming-Yee Chiu, comments in "Is Industry Ready for Machine Vision?" panel discussion, *Machine Vision for Inspection and Measurement*, edited by Herbert Freeman, Academic Press: New York, 1989, pp. 223-236.

[36] Alok N. Choudhary and Janak H. Patel, *Parallel Architectures and Parallel Algorithms for Intergrated Vision Systems*, Kluwer Academic: Boston, 1990.

[37] Chen-Chau Chu, N. Nandhakumar, and J. K. Aggarwal, "Image Segmentation Using Laser Radar Data," *Pattern Recognition*, Vol. 23, 1990, pp. 569-581.

[38] James Coggins and Anil K. Jain, "A Spatial Filtering Approach to Texture Analysis," *Pattern Recognition Letters*, Vol. 3, 1985, pp. 195-203.

[39] Jonathan Courtney, *Mobile Robot Localization Using Pattern Classification Techniques*, Master's Thesis, Michigan State University Department of Computer Science, 1993.

[40] O. Dahle, A. Sommerfelt, and A. McLeod, "On-Line Inspection of Extruded Profile Geometry," *Proceedings of SME VISION '90*, Detroit, November 1990, pp. 6-1 to 6-9.

[41] Wayne Daley and Tandoni Rao, "Color Vision for Industrial Inspection," *Proceedings of SME VISION '90*, Detroit, November 1990, pp. 12-11 to 12-24.

[42] Wayne Daley and Richard A. Carey, "Color Machine Vision for Defect Detection (Algorithms and Techniques)," *Proceedings of International Robots and Vision Automation Conference*, October 1991, Detroit, pp. 5-29 to 5-42.

[43] Z. Dong and A. Soom, "Methods for Automating Tolerance Analysis and Synthesis in CAD Environments," *Proceedings of the Third International Symposium on Robotics and Manufacturing: Research, Education, and Applications (ISRAM '90)*, Burnaby, BC, July 18-20, 1990, pp. 791-796.

[44] Richard C. Dorf, editor, *International Encyclopedia of Robotics: Applications and Automation*, John Wiley & Sons: New York, 1988.

[45] David D. Dreyfus, comments in "Is Industry Ready for Machine Vision?" panel discussion, *Machine Vision for Inspection and Measurement*, edited by Herbert Freeman, Academic Press: New York, 1989, pp. 223-236.

[46] Steven Dubowsky, Krisztina Holly, Annie L. Murray, and Joseph M. Wander, "Design Optimization of Moiré Interferometers for Rapid 3-D Manufacturing Inspection," *Proceedings of the Machine Vision Systems Integration in Industry Conference, SPIE Vol. 1386*, Boston, 1990, pp. 10-20.

[47] Richard O. Duda, David Nitzan, and Phyllis Barrett, "Use of Range and Reflectance Data to Find Planar Surface Regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, 1979, pp. 259-271.

[48] K. Edamatsu, A. Komuro, and Y. Nitta, "Application of Random Pattern Recognition Technique to Quantitative Evaluation of Automatic Visual Inspection Algorithms," *Proceedings of the IEEE Workshop on Industrial Applications of Machine Vision*, Research Triangle Park, North Carolina, 1982, pp. 139-143.

[49] O. D. Faugeras and M. Hebert, "The Representation, Recognition, and Locating of 3-D Objects," *The International Journal of Robotics Research*, Vol. 5, No. 3, Fall 1986, pp. 27-52.

[50] O. D. Faugeras, M. Hebert, and E. Pauchon, "Segmentation of Range Data into Planar and Quadratic Patches," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, 1983, pp. 8-13.

[51] T. Fan, G. Medioni, and R. Nevatia, "Recognizing 3D Objects Using Surface Descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, 1989, pp. 1140-1157.

[52] F. P. Ferrie, J. Lagarde, and P. Whaite, "Darboux Frames, Snakes, and Super-Quadrics: Geometry from the Bottom-Up," *Proceedings of the IEEE Workshop on the Interpretation of 3D Scenes*, Austin, 1989, 170-176.

[53] Martin A. Fischler and Robert C. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, 1981, pp. 381-395.

[54] Robert B. Fisher, "Non-Wildcard Matching Beats Interpretation Tree," *Proceedings of the 1992 British Machine Vision Conference*, Leeds, U.K., September 1992, pp. 560-569.

[55] Patrick J. Flynn, *CAD-Based Computer Vision: Modeling and Recognition Strategies*, Ph.D. Dissertation, Michigan State University, Department of Computer Science, 1990.

[56] Patrick J. Flynn and Anil K. Jain, "Surface Classification: Hypothesis Testing and Parameter Estimation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, 1988, 261-267.

[57] Patrick J. Flynn and Anil K. Jain, "On Reliable Curvature Estimation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, 1989, 110-116.

[58] Patrick J. Flynn and Anil K. Jain, "BONSAI: 3-D Object Recognition Using Constrained Search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, pp. 1066-1075.

[59] Patrick J. Flynn and Anil K. Jain, "CAD-Based Computer Vision: From CAD Models to Relational Graph," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, pp. 114-132.

[60] Herbert Freeman, editor, *Machine Vision for Inspection and Measurement: proceedings of the second annual workshop sponsored by the Center for Computer Aids for Industrial Productivity at Rutgers University on April 25-26, 1988*, Academic Press: New York, 1989.

[61] Michael Gay, "Segmentation Using Region Merging with Edges," *Proceedings of the Fifth Alvey Vision Conference*, Reading University, U.K., September 1989, pp. 115-119.

[62] K. Goebbels and G. Ferrano, "Automation of Surface Defect Detection and Evaluation," *Proceedings of the Conference on Automated Inspection and High Speed Vision Architectures, SPIE Vol. 849*, Cambridge, Massachusetts, November 1987, pp. 117-124.

[63] Istvan Gorog, comments in "Is Industry Ready for Machine Vision?" panel discussion, *Machine Vision for Inspection and Measurement*, edited by Herbert Freeman, Academic Press: New York, 1989, pp. 223-236.

[64] W. Eric L. Grimson, "The Combinatorics of Heuristic Search Termination for Object Recognition in Cluttered Environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, pp. 920-935.

[65] W. Eric L. Grimson and Tomás Lozano-Peréz, "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *International Journal of Robotics Research*, Vol. 3, No. 3, 1984, pp. 3-35.

[66] Kristjan T. Gunnarsson and Friedrich B. Prinz, "CAD Model-Based Localization of Parts in Manufacturing," *Computer*, Vol. 20, No. 8, 1987, pp. 66-74.

[67] J. Gustafson, "Reevaluating Amdahl's Law," *Communications of the ACM*, Vol. 31, 1988, pp. 523-533.

[68] Ernest L. Hall, James K. B. Tio, Charles A. McPherson, and Firooz A. Sadjadi, "Measuring Curved Surfaces for Robot Vision," *IEEE Computer*, Vol. 15, 1982, pp. 42-54.

[69] Charles D. Hansen and Thomas C. Henderson, "CAGD-Based Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, 1989, pp. 1181-1193.

[70] Robert Haralick and Linda Shapiro, *Computer and Robot Vision Volume II*, Addison-Wesley: New York, 1993.

[71] Martial Hebert and Jean Ponce, "A New Method for Segmenting 3-D Scenes Into Primitives," *Proceedings of the Sixth International Conference on Pattern Recognition*, Munich, 1982, 836-838.

[72] Kristina Hedengren, "Automatic Image Analysis for Inspection of Complex Objects: a Systems Approach," *Proceedings of the Eighth International Conference on Pattern Recognition*, Paris, 1986, pp. 919-921.

[73] Kristina Hedengren, "Methodology for Automatic Image-Based Inspection of Industrial Objects," in J. Sanz (ed.), *Advances in Machine Vision*, Springer-Verlag: New York, 1989, pp. 160-191.

[74] D. A. Hill, "The Context of Automated Inspection," in Batchelor et al. (eds.), *Automated Visual Inspection*, IFS Publications Ltd.: Kempston, Bedford, England, 1985, pp. 1-8.

[75] R. Hoffman and A. K. Jain, "Segmentation and Classification of Range Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, 1987, pp. 608-620.

[76] "Hommel Surface Roughness: Terminology and Parameters," brochure, Hommel America Company, New Britain, Connecticut.

[77] Tsai-Hong Hong and Michael O. Shneier, "Rotation and Translation of Objects Represented by Octrees," *Proceedings of the 1987 International Conference on Robotics and Automation*, Raleigh, NC, April 1987, pp. 947-952.

[78] Berthold K. P. Horn, *Robot Vision*, MIT Press: Cambridge, MA, 1986.

[79] G. Hu and G. C. Stockman, "3-D Surface Solution Using Structured Light and Constraint Propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 11, 1989, pp. 390-402.

[80] Robert Huber, General Motors Saginaw Malleable Iron Plant, personal communication, January 3, 1993.

[81] V. Daniel Hunt, *Smart Robots: A Handbook of Intelligent Robotic Systems*, Chapman and Hall: New York, 1985.

[82] Daniel P. Huttenlocher and W. J. Rucklidge, "A Multi-Resolution Technique for Comparing Images Using the Hausdorff Distance," Cornell University Department of Computer Science Technical Report, December 1992.

[83] K. Ikeuchi and T. Kanade, "Automatic Generation of Object Recognition Programs," *Proceedings of the IEEE,* Vol. 76, 1988, pp. 1016-1035.

[84] Kenichi Isoda and Yasuo Nakagawa, "Inspection Robot: Advanced Robotic Inspection," in Dorf (ed.), *International Encyclopedia of Robotics: Applications and Automation*, John Wiley & Sons: New York, 1988, pp. 668-679.

[85] Anil K. Jain, "Object Recognition Using Range Images," *Proceedings of the Eighth Scandinavian Conference on Image Analysis*, Tromso, Norway, May 1993, pp. 797-805.

[86] Anil K. Jain and Richard C. Dubes, *Algorithms for Clustering Data*, Prentice Hall: Englewood Cliffs, N.J., 1988.

[87] Anil K. Jain and Marie-Pierre Dubuisson, "Segmentation of X-Ray and C-Scan Images of Fiber Reinforced Composite Materials," *Pattern Recognition*, Vol. 25, No. 3, 1992, pp. 257-270.

[88] Anil K. Jain and Patrick J. Flynn, editors, *Three-Dimensional Object Recognition Systems*, Elsevier: New York, 1993.

[89] Anil K. Jain and Richard Hoffman, "Evidence-Based Recognition of 3-D Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, 1988, pp. 783-802.

[90] Anil K. Jain and Sateesha G. Nadabar, "MRF Model-Based Segmentation of Range Images," *Proceedings of the Third IEEE International Conference on Computer Vision*, Osaka, Japan, 1990, pp. 667-671.

[91] Anil K. Jain, Tim Newman, and Michael Goulish, "Range-intensity Histogram for Segmenting LADAR Images," *Pattern Recognition Letters*, Vol. 13, 1992, pp. 41-56.

[92] Ramesh C. Jain and Anil K. Jain, editors, *Analysis and Interpretation of Range Images*, Springer-Verlag: New York, 1990.

[93] I. T. Joliffe, *Principle Component Analysis*, Springer-Verlag: Berlin, 1986.

[94] James Just, Keith King, Michael Osheroff, George Berke, Peter Spidaliere, and Tran Ngoc, "Robotic Technology: An Assessment and Forecast," in Ayres, et al. (eds.), *Robotics and Flexible Manufacturing Technologies*, Noyes Publications: Park Ridge, NJ, 1985.

[95] N. P. Juster, "Modelling and Representation of Dimensions and Tolerances: A Survey," *Computer-Aided Design*, Vol. 24, No. 1, 1992, pp. 3-17.

[96] Michael Kassler, Peter I. Corke, and Paul C. Wong, "Automatic Grading and Packing of Prawns," *Proceedings of the International Robots and Vision Automation Conference*, Detroit, MI, 1991, pp. 9-27 to 9-32.

[97] Clifford W. Kennedy, Edward G. Hoffman, and Steven D. Bond, *Inspection and Gaging, Sixth Edition*, Industrial Press, Inc.: New York, 1987.

[98] Jun-Ing Ker and Khokiat Kengskool, "An Efficient Method for Inspecting Machined Parts by a Fixtureless Machine Vision System," *Proceedings of SME VISION '90*, Detroit, November 1990, pp. 2-45 to 2-51.

[99] Jerry Kirsch and Kerry Kirsch, "Inspection Robot: Applications in Industry," in *Concise International Encyclopedia of Robotics*, Richard C. Dorf (ed.), John Wiley & Sons: New York, 1990, pp. 417-422.

[100] J. J. Koenderink and A. J. van Doorn, "The Internal Representation of Solid Shape with Respect to Vision," *Biological Cybernetics*, Vol. 32, 1979, pp. 211-216.

[101] Stanley N. Lapidus, "New Techniques for Industrial Vision Inspection," *Proceedings of the Third Annual Applied Machine Vision Conference*, Schaumburg, Illinois, 1984, pp. 4-1 to 4-12.

[102] Mark H. Lawry, *I-DEAS Student Guide*, Structural Dynamics Research Corporation, Milford, OH, 1991.

[103] Greg C. Lee and George C. Stockman, "Detecting Object Wings in Quadric Surface Scenes," *Proceedings of the Conference on Applications of Artificial Intelligence X: Machine Vision and Robotics, SPIE Vol. 1708*, Orlando, FL, April 1992, pp. 335-344.

[104] Mark H. Lee, *Intelligent Robotics*, Open University Press: Milton Keynes, U.K., 1989.

[105] Xiaobo Li and Richard C. Dubes, "The First Stage in Two-Stage Template Matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 7, 1985, pp. 700-707.

[106] James C. Lin, Wen-Hsiang Tsai, Jeunn-Shenn Lee, and Chai-Hsiung Chen, "A Vision System with Automatic Learning Capability for Industrial Parts Inspection," *Proceedings of the IEEE International Conference on Robotics*, Atlanta, 1984, pp. 417-425.

[107] T. I. Liu, "Automated Visual Inspection of Drill Wear," *Proceedings of Manufacturing International '90, Vol. V, Design, Reliability, and Education of Manufacturability*, Atlanta, 1990, pp. 115-119.

[108] Tomàs Lozano-Perèz, W. Eric L. Grimson, and Steven J. White, "Finding Cylinders in Range Data," *Proceedings of the 1987 IEEE International Conference on Robotics & Automation*, Raleigh, N.C., 1987, pp. 202-207.

[109] Gordon M. Mair, *Industrial Robotics*, Prentice Hall: New York, 1988.

[110] Andrew David Marshall, *The Automatic Inspection of Machined Parts Using Three-Dimensional Range Data and Model-Based Matching Techniques*, Ph.D. Dissertation, Department of Computing Mathematics, University of Wales College of Cardiff, 1989.

[111] Andrew David Marshall, "Automatically Inspecting Gross Features of Machined Objects Using Three-Dimensional Depth Data," *Proceedings of the SPIE Conference on Machine Systems Integration in Industry Vol. 1386*, Boston, 1990, pp. 243-254.

[112] Andrew David Marshall and Ralph R. Martin, *Computer Vision, Models, and Inspection*, World Scientific: River Edge, NJ, 1992.

[113] Susan Lloyd McGarry, "Acceptable Quality Levels are not Acceptable any more," *Proceedings of Autofact 6*, Anaheim, California, 1984, pp. 15-1 to 15-13.

[114] Francis L. Merat, Gerald M. Radack, and Jeremy F. Jacobsohn, "Automatic Inspection Planning within a Feature-Based CAD System," *Proceedings of the Third International Symposium on Robotics and Manufacturing: Research, Education, and Applications (ISRAM '90)*, Burnaby, BC, July 18-20, 1990, pp. 723-729.

[115] James R. Miller, "Sculptured Surfaces in Solid Models: Issues and Alternative Approaches," *IEEE Computer Graphics and Applications*, Vol. 6, No. 12, 1986, pp. 37-48.

[116] James R. Miller, "Architectural Issues in Solid Modelers," *IEEE Computer Graphics & Applications*, Vol. 9, No. 9, September 1989, pp. 72-87.

[117] Steven Miller, "Recent Developments in Robotics and Flexible Manufacturing Systems," in Ayres, et al. (eds.), *Robotics and Flexible Manufacturing Technologies*, Noyes Publications: Park Ridge, NJ, 1985.

[118] Richard S. Millman and George D. Parker, *Elements of Differential Geometry*, Prentice-Hall: Englewood Cliffs, NJ, 1977.

[119] Claudio Mirolo and Enrico Pagello, "A Solid Modeling System for Robot Action Planning," *IEEE Computer Graphics and Applications*, Vol. 9, No. 1, January 1989, pp. 55-69.

[120] Amar Mitiche and J.K. Aggarwal, "Multiple Sensor Integration/Fusion through Image Processing: a Review," *Optical Engineering*, Vol. 25, 1986, pp. 380-386.

[121] Yolande Muller and Roger Mohr, "Planes and Quadrics Detection Using Hough Transform," *Proceedings of the Seventh International Conference on Pattern Recognition*, Montreal, 1984, pp. 1101-1103.

[122] Sateesha G. Nadabar and Anil K. Jain, "Edge Detection and Labeling by Fusion of Intensity and Range Images," *Proceedings of the Conference on Applications of AI: Machine Vision and Robotics, SPIE Vol. 1708*, Orlando, FL, April 1992, pp. 108-119.

[123] Tadashi Nagata and Hong-Bin Zha, "Recognizing and Locating a Known Object from Multiple Images," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, 1991, pp. 434-448.

[124] Shree Nayar, Lee Weiss, David Simon, and Arthur Sanderson, "Structural Highlight Inspection of Specular Surfaces Using Extended Gaussian Images," *1988 Carnegie Mellon Robotics Institute Annual Research Review*, pp. 55-65.

[125] Timothy S. Newman, Richard J. Enbody, and Anil K. Jain, "3D Object Recognition: Interpretation Tree Search on a MIMD Machine," *Proceedings of Eleventh International Conference on Pattern Recognition*, the Hague, the Netherlands, August 1992, pp. A-337 to A-340.

[126] Timothy S. Newman, Patrick J. Flynn, and Anil K. Jain, "Model-Based Surface Classification," *Proceedings of the SPIE Conference on Geometric Methods in Computer Vision Vol. 1570*, San Diego, July 1991, pp. 250-261.

[127] Timothy S. Newman, Patrick J. Flynn, and Anil K. Jain, "Model-Based Classification of Quadric Surfaces," to appear in *Computer Vision, Graphics, and Image Processing: Image Understanding*, 1993.

[128] Timothy S. Newman, Anil K. Jain, and H. R. Keshavan, "3D CAD-Based Inspection I: Coarse Verification," *Proceedings of Eleventh International Conference on Pattern Recognition*, the Hague, the Netherlands, August 1992, pp. A-49 to A-52.

[129] Takanori Ninomiya, Mineo Nomoto, and Yasuo Nakagawa, "Automatic $2\frac{1}{2}$D Shape Inspection System for Via-Hole Fillings of Green Sheets by Shadow Image Analysis," *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, Arizona, 1989, pp. 515-520.

[130] David Nitzan, "Three-Dimensional Vision Structure for Robot Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, 1988, pp. 291-309.

[131] David Nitzan, Alfred E. Brain, and Richard O. Duda, "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis," *Proceedings of the IEEE*, Vol. 65, 1977, pp. 206-220.

[132] A. Noble, V. D. Nguyen, C. Marinos, A. T. Tran, J. Farley, K. Hedengren, and J. L. Mundy, "Template Guided Visual Inspection," *Proceedings of the Second European Conference on Computer Vision*, Santa Margherita Ligure, Italy, May 1992, pp. 893-901.

[133] Gotaro Odawara, Ichiro Ogata, Masayuki Arase, Hiroaki Shirakawa, and Hiroshi Takeya, "An Integrated Pattern Inspection System," *Proceedings of the Eighth International Conference on Pattern Recognition*, Paris, 1986, pp. 5-8.

[134] S. Olympieff, J. C. Pineda, and P. Horaud, "The use of Vision for Industrial Inspection," *Proceedings of the Sixth International Conference on Automated Inspection and Product Control*, Birmingham, U.K., 1982, pp. 179-190.

[135] Sharath Pankanti, Chitra Dorai, and Anil K. Jain, "Robust Feature Detection for 3D Object Recognition and Matching," *Proceedings of the SPIE Conference on Geometric Methods in Computer Vision II*, San Diego, July 1993.

[136] H. D. Park and O. R. Mitchell, "CAD Based Planning and Execution of Inspection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, 1988, pp. 858-863.

[137] Jong-Seok Park and Julius T. Tou, "A Solder Joint Inspection System for Automated Printed Circuit Board Manufacturing," *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, 1990, pp. 1290-1295.

[138] Srinivasan Parthasarathy, Denise Wolf, Evelyn Hu, Susan Hackwood, and Gerardo Beni, "A Color Vision System for Film Thickness Determination," *Proceedings of the IEEE International Conference on Robotics and Automation*, Raleigh, North Carolina, 1987, pp. 515-519.

[139] D. Petkovic, J. Sanz, K. Mohiuddin, E. Hinkle, M. Flickner, C. Cox, and K. Wong, "An Experimental System for Disk Head Inspection," *Proceedings of the Eighth International Conference on Pattern Recognition*, Paris, 1986, pp. 9-13.

[140] Les Piegl, "On NURBs: A Survey," *IEEE Computer Graphics and Applications*, Vol. 11, No. 1, 1991, pp. 55-71.

[141] Harry H. Poole, *Fundamentals of Robotics Engineering*, Van Nostrand Reinhold: New York, 1989.

[142] Tim Pryor, "Optical Inspection and Machine Vision," *Proceedings of the IEEE Workshop on Industrial Applications of Machine Vision*, Research Triangle Park, North Carolina, 1982, pp. 3-20.

[143] Karl Ratcliff, Alun Harris, Charles Allen, and Robert Bicker, "3-Dimensional Datuming and Inspection of Mechanical Parts using Wire-frame Modeling," *Proceedings of the Automated Inspection and High-Speed Vision Architectures III Conference, SPIE Vol. 1197*, Philadelphia, 1989, pp. 230-236.

[144] G. R. Reid, R. C. Rixon, and S. J. Marshall, "3-D Machine Vision for Automatic Measurements of Complex Shapes," *Proceedings of the Seventh International Conference on Automated Inspection and Product Control*, Birmingham, U.K., 1985, pp. 129-138.

[145] Aristides A. G. Requicha, "Representation of Tolerances in Solid Modeling: Issues and Alternative Approaches," in *Solid Modeling by Computers*, ed. by M. S. Pickett and J. W. Boyse, Plenum Press: New York, 1984.

[146] Aristides A. G. Requicha and Stephen C. Chan, "Representation of Geometric Features, Tolerances, and Attributes in Solid Modelers Based on Constructive Geometry," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 3, September 1986, pp. 156-166.

[147] Earl H. Rideout, "Lowering Test Costs with 3D Solder-Joint Inspection," *Test and Measurement World*, January 1990, pp. 61-66.

[148] C. A. Rosen, "Machine Vision and Robotics: Industrial Requirements," in *Computer Vision and Sensor-Based Robots*, George Dodd and Lothar Rossol (eds.), Plenum Press: New York, 1979, pp. 3-18.

[149] Azriel Rosenfeld and Gordon J. Vanderbrug, "Coarse-Fine Template Matching," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 7, No. 2, 1977, pp. 104-107.

[150] Lothar Rossol, "Computer Vision in Industry," in *Robot Vision*, Alan Pugh (ed.), IFS Publications: Kempston, Bedford, England, 1983, pp. 11-18.

[151] Hanan Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley: New York, 1990.

[152] Jorge Sanz, editor, *Advances in Machine Vision*, Springer-Verlag: New York, 1989.

[153] Richard Q. Schmidt, "Inspection and adaptive Robot Applications based on Three-dimensional Vision Measurements," *Proceedings of the SPIE Intelligent Robots and Computer Vision Conference Vol. 521*, Cambridge, Massachusetts, 1984, pp. 346-351.

[154] Richard Q. Schmidt, "Three Dimensional Vision applied to adaptive Robotic Control and Inspection," *Proceedings of the Second Annual International Robot Conference*, Long Beach, California, 1984, pp. 60-65.

[155] Yoshiaki Shirai, K. Koshikawa, M. Oshima, and K. Ikeuchi, "A Vision System Based on Three-Dimensional Model," *Proceedings of the International Conference on Advanced Robotics*, Tokyo, September 1983, pp. 139-146.

[156] Olli Silvén, Ilkka Virtanen, and Matti Pietikainen, "CAD Data-based Comparison Method for Printed Wiring Board (PWB) Inspection," *Proceedings of the SPIE Intelligent Robots and Computer Vision Conference Vol. 521*, Cambridge, Massachusetts, 1984, pp. 400-405.

[157] Olli Silvén, Tapani Westman, Seppo Huotari, and Hannu Hakalahti, "A Defect Analysis Method for Visual Inspection," *Proceedings of the Eighth International Conference on Pattern Recognition*, Paris, 1986, pp. 868-870.

[158] Olli Silvén, Ilkka Virtanen, Tapani Westman, Timo Piironen, and Matti Pietikainen, "A Design Data-Based Visual Inspection System for Printed Wiring," in J. Sanz (ed.), *Advances in Machine Vision*, Springer-Verlag: New York, 1989, pp. 192-212.

[159] Frank L. Skaggs, "Vision-based Parts Measurement System," *Proceedings of the Third International Conference on Robot Vision and Sensory Controls*, Cambridge, Massachusetts, 1983, pp. 579-589.

[160] K. K. Sreenivasan, M. D. Srinath, and A. R. Khotanzad, "Automated Vision System for Inspection of Wedge Bonds," *Proceedings of the Conference on Machine Vision Applications in Character Recognition and Industrial Inspection, SPIE Vol. 1661*, San Jose, CA, 1992, pp. 425-430.

[161] Charles N. Stevenson, "Model-Based Programming and Control of Robot Manipulators,"*Computer*, Vol. 20, No. 8, August 1987, pp. 76-84.

[162] George C. Stockman, "Object Recognition and Localization via Pose Clustering," *Computer Vision, Graphics, and Image Processing*, Vol. 40, 1987, pp. 361-387.

[163] Steve Sullivan, Lorraine Sandford, and Jean Ponce, "On Using Geometric Distance Fits to Estimate 3D Object Shape, Pose, and Deformation from Range, CT, and Video Images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, June 1993, pp. 110-115.

[164] Bindinganavle R. Suresh, Richard A. Fundakowski, Tod S. Levitt, and John Overland, "A Real-Time Automated Visual Inspection System for Hot Steel Slabs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, 1983, pp. 563-572.

[165] Joseph Szafranski, Leonard H. Bieman, and Kevin G. Harding, "CADEYES – A New 3D Gaging System," *Proceedings of the International Robots and Vision Automation Conference*, Detroit, October 1991, pp. 12-67 to 12-79.

[166] Glenn Tarbox and Lester Gerhardt, "Design and Implementation of a Hierarchical Automated Inspection System," *Proceedings of the SPIE Automated Inspection and High-Speed Vision Architectures III Conference Vol. 1197*, Philadelphia, 1989, pp. 13-24.

[167] Gabriel Taubin, "Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, pp. 1115-1138.

[168] Russel W. Tayler, "Improved Surface Extraction Via Parameter Space Voting Techniques," *Proceedings of the Conference on Curves and Surfaces in Computer Vision and Graphics, SPIE Vol. 1251*, Santa Clara, California, 1990, pp. 380-391

[169] Warren Taylor, "Developmental Inspection Programs for Medical Products," *Proceedings of SME VISION '90*, Detroit, November 1990, pp. 3-1 to 3-10.

[170] Technical Arts Corporation, *100X User's Manual and Application Programming Guide*.

[171] Wayne Tiller, "Rational B-Splines for Curve and Surface Representation," *IEEE Computer Graphics and Applications*, Vol. 3, No. 6, September 1983, pp. 61-69.

[172] Carme Torras, editor, *Computer Vision: Theory and Industrial Applications*, Springer-Verlag: New York, 1992.

[173] H. Tropf, H. Geisselmann, and J. P. Foith, "Some Applications of the Fast Industrial Vision System S.A.M.," *Proceedings of the IEEE Workshop on Industrial Applications of Machine Vision*, Research Triangle Park, North Carolina, 1982, pp. 73-79.

[174] E. Trucco and A. M. Wallace, "Draft Technical Report for IED LAIRD Project: Planning Sensor Positions for Feature Inspections," Heriot Watt University, 1992.

[175] Costas Tsatsoulis and King-Sun Fu, "A Computer Vision System for Assembly Inspection," *Proceedings of the SPIE Intelligent Robots and Computer Vision Conference Vol. 521*, Cambridge, Massachusetts, 1984, pp. 352-357.

[176] Joshua U. Turner and Michael J. Wozny, "A Mathematical Theory of Tolerances," *Geometric Modeling for CAD Applications: Selected and Expanded Papers from the IFIP WG 5.2 Working Conference, Rensselaerville, NY, 1986*, ed. by M. J. Wozny, H. W. McLaughlin, and J. L. Encarnaao, Elsevier Science: New York, 1988, pp. 163-187.

[177] L. Vanderheydt, A. Oosterlinck, and H. Van den Berghe, "Experiments of Computer Vision Techniques for Industrial Applications," *Proceedings of the IEEE Workshop on Industrial Applications of Machine Vision*, Research Triangle Park, North Carolina, 1982, pp. 21-25.

[178] L. Van Gool, P. Wambacq, and A. Oosterlinck, "Intelligent Robotic Vision Systems," in *Intelligent Robotic Systems*, edited by Spyros G. Tzafestas, Marcel Dekker: New York, 1991, pp. 457-507.

[179] J. G. Verly, R. L. Delanoy, and D. E. Dudgeon, "Machine Intelligence Technology for Automatic Target Recognition," *The Lincoln Laboratory Journal*, Vol. 2, 1989, pp. 277-307.

[180] Friedrich M. Wahl and Hans-Peter Biland, "Decomposition of Polyhedral Scenes in Hough Space," *Proceedings of the Eighth International Conference on Pattern Recognition*, Paris, 1986, pp. 78-84.

[181] Stephen Walsh, *Indoor Robot Navigation using a Symbolic Landmark Map*, Ph.D. Thesis, Michigan State University Department of Computer Science, 1992.

[182] Charles C. Weems, Christopher Brown, Jon A. Webb, Tomaso Poggio, and John R. Kender, "Parallel Processing in the DARPA Strategic Computing Vision Program," *IEEE Expert*, Vol. 6, No. 5, October 1991, pp. 23-38.

[183] Juyang Weng, Paul Cohen, and Nicolas Rebibo, "Motion and Structure Estimation from Stereo Image Sequences," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, 1992, pp. 362-382.

[184] Geoff A. W. West, Terrence Fernando, and Peter M. Dew, "CAD Based Inspection: Using a Vision Cell Demonstrator," *Proceedings of the IEEE Workshop on Directions in Automated CAD-Based Vision*, Maui, HA, June 1991, pp. 155-164.

[185] G. Barrie Wetherill, *Sampling Inspection and Quality Control*, Methuen and Company, Ltd.: London, 1969.

[186] G. Winkler, "Automated Visual Inspection," *Proceedings of the Industrial Applications of Image Analysis Workshop*, Antwerp, 1983, pp. 13-30.

[187] Paul Kenneth Wright and David Alan Bourne, *Manufacturing Intelligence*, Addison-Wesley: New York, 1988.

[188] Xuanli Lisa Xie and Gerardo Beni, "A Validity Measure for Fuzzy Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, pp. 841-847.

[189] Lei Xu, Erkki Oja, and Pekka Kultanen, "A new Curve Detection Method: Randomized Hough Transform (RHT)," *Pattern Recognition Letters*, Vol. 11, 1990, pp. 331-338.

[190] H. S. Yang and Avi C. Kak, "Determination of the Identity, Position, and Orientation of the Topmost Object in a Pile," *Proceedings of the Third IEEE Workshop on Computer Vision: Representation and Control*, Bellaire, MI, 1985, pp. 38-48.

[191] Qin-Zhong Ye and Per E. Danielsson, "Visual Inspection of Printed Circuit Boards Using Connectivity Preserving Shrinking," *Proceedings of the Eighth International Conference on Pattern Recognition*, Paris, 1986, pp. 2-4.

[192] Naokazu Yokoya and Martin D. Levine, "Range Image Segmentation Based on Differential Geometry: A Hybrid Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 6, 1989, pp. 643-649.

[193] Naokazu Yokoya and Martin D. Levine, "Volumetric Description of Solids of Revolution in a Range Image," *Proceedings of the Tenth International Conference on Pattern Recognition*, Atlantic City, June 1990, pp. 303-309.

[194] Nello Zuech, *Applying Machine Vision*, John Wiley & Sons: New York, 1988.