# Automatic Reconstruction of Industrial Installations Using Point Clouds and Images

Tahir Rabbani Shah

Publications on Geodesy 62

**NCG** Nederlandse Commissie voor Geodesie Netherlands Geodetic Commission

Delft, May 2006

Automatic Reconstruction of Industrial Installations Using Point Clouds and Images
Tahir Rabbani Shah
Publications on Geodesy 62
ISBN-10: 90 6132 297 9
ISBN-13: 978 90 6132 297 9
ISSN: 0165 1706

Published by: NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission,
Delft, The Netherlands.
Printed by: Optima Grafische Communicatie, Optima Graphic Communication, Rotterdam,
The Netherlands.
Cover illustration: Fitting of a model to a point cloud (left). Fitting a model to an image (right).

NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission
P.O. Box 5058, 2600 GB Delft, The Netherlands
T: +31 (0)15 278 28 19
F: +31 (0)15 278 17 75
E: info@ncg.knaw.nl
W: www.ncg.knaw.nl

The NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission is part of
the Royal Netherlands Academy of Arts and Sciences (KNAW).

# Abstract

## Automatic reconstruction of industrial installations using point clouds and images

### Tahir Rabbani Shah

Up to date and accurate 3D models of industrial sites are required for different applications like planning, documentation and training. Traditional methods for acquiring as-built information like manual measurements by tape and tacheometry are not only slow and cumbersome but most of the time they also fail to provide the amount of detail required. Many industrial facilities provide a limited personnel access because of the presence of radioactive, toxic or hazardous materials together with an unsafe working environment, which necessitates the use of non-contact measurement methods.

Traditional photogrammetry depends on point or line measurements from which it is very hard to get complete CAD models without extensive manual editing and refinement. Compared to photogrammetry laser scanning provides explicit and dense 3D measurements. There has been a rapid increase in the speed and accuracy of the laser scanners in the last decade, while their costs and sizes have been continuously shrinking. All modeling tools available on the market depend on heavy operator intervention for most of the modeling tasks. Although there are some semi-automatic tools like plane or cylinder growing even there the operator has to start the growing process for each primitive. Furthermore, the fitted surfaces must be manually edited by the operator to convert them to a CAD description.

This thesis presents new methods and techniques which can be used for automatic or efficient semiautomatic 3D modeling of existing industrial installations from point clouds and images. The goal is to use explicit 3D information from the point clouds to automatically detect the objects and structure present in the scene. The detected objects are then used as targets for model based registration, which can be automated by searching for object correspondences. To avoid manual editing the presented techniques use models from a catalogue of commonly found CAD objects as templates for model fitting. In the final fitting phase images are also included to improve the quality of parameter estimation.

Segmentation is a very important step that needs to be carried out as a pre cursor to object recognition and model fitting. We present a method for the segmentation of the point clouds, which avoids over-segmentation while partitioning the input data into mutually disjoint, smoothly connected regions. It uses a criterion based on a combination of surface normal similarity and spatial connectivity, which we call smoothness constraint. As we do not use surface curvature our algorithm is less sensitive to noise. Moreover, there are only a few parameters which can be adjusted to get a desired trade-off between under- and over-segmentation.

Segmentation is followed by a stage of object recognition based on a variation of the Hough transform for automatic plane and cylinder detection in the point clouds. For plane detection the Hough transform is three dimensional. For the cylinder detection the direct application of the Hough transform requires a 5D Hough space, which is quite impractical because of its space and computational complexity. To resolve this problem we present a two-step approach requiring a 2D and 3D Hough space. In the first step we detect strong hypotheses for the cylinder orientation. The second step estimates the remaining three parameters of the cylinder i.e. radius and position.

The problem of fitting models like planes, cylinders, spheres, cones, tori and CSG models to point clouds is very important for data reduction. For the fitting of CSG models this thesis presents three different methods for approximating the orthogonal distance, which are compared based on speed and accuracy.

We also present methods for using modeled objects in individual scans as targets for registration. As the available geometric structure is used, there is no need to place artificial targets. We present two different methods for this purpose called Indirect and Direct method. The Indirect method is a quick way to get approximate values while the Direct method is then used to refine the approximate solution. We also present techniques for automatically finding the corresponding objects for registration of scans. The presented techniques are based on constraint propagation which use the geometric information available from the previously made correspondence decision to filter out the possibilities for future correspondences.

Although point clouds are very important for the automation because of their explicit 3D information, images provide a complementary source of information as they contain well-defined edges of the bounded objects. We present methods for the fitting of CSG models to a combination of point clouds and images. We also present techniques for the specification of geometric constraints between sub-parts of a CSG tree and their inclusion in the model estimation process. A taxonomy of commonly encountered geometric constraints and their mathematical formulation is also given.

We hope that the techniques presented in this thesis will lead to an improvement in efficiency and quality of the models obtained for industrial installations from point clouds and images.

# Samenvatting

## Automatische reconstructie van industriële installaties uit puntenwolken en beelden

### Tahir Rabbani Shah

Actuele en precieze 3D modellen van industriële installaties zijn nodig voor een verscheidenheid aan toepassingen zoals planning, documentatie en training. Traditionele methoden voor de inwinning van as-built informatie - bijvoorbeeld met behulp van een meetband en tachymetrie - zijn niet alleen tijdrovend en moeizaam, maar zijn meestal ook niet in staat om de benodigde hoeveelheid detail te leveren. Veel industriële complexen zijn beperkt toegankelijk als gevolg van radioactieve, toxische of anderszins gevaarlijke stoffen. Een onveilige werkomgeving maakt het gebruik van contactloze meetmethoden noodzakelijk.

Traditionele fotogrammetrie is afhankelijk van punt- of lijnmetingen waarmee het erg moeilijk is om complete CAD-modellen te vervaardigen zonder veel handmatig werk. In tegenstelling tot fotogrammetrie levert laserscanning expliciete 3D metingen met een hoge dichtheid. In de laatste tien jaar zijn de laserscanners aanzienlijk sneller en preciezer geworden, terwijl de kosten en de grootte steeds afnemen. Alle op de markt beschikbare modelleersoftware vraagt veel interactie van een operateur. Wel zijn er enkele semi-automatische hulpmiddelen beschikbaar zoals voor het automatisch ”laten groeien” van vlakken of cilinders, maar zelfs daar moet de operateur het groeiproces voor ieder object starten. Bovendien moeten de gevonden oppervlakken handmatig bewerkt worden om ze te converteren naar een CAD-beschrijving.

Dit proefschrift presenteert nieuwe methoden en technieken die gebruikt kunnen worden voor automatische of efficiënte semi-automatische 3D-modellering van bestaande industriële installaties met behulp van puntenwolken en beelden.

Het doel is om de expliciete 3D-informatie van de puntenwolken te gebruiken om automatisch objecten en structuur in de scne te vinden. De gevonden objecten worden vervolgens gebruikt als referentieobjecten voor modelgebaseerde registratie, die geautomatiseerd kan worden door te zoeken naar corresponderende objecten. Om handwerk te vermijden gebruiken de gepresenteerde technieken modellen uit een catalogus met veel voorkomende CAD-objecten als mallen voor modelfitting. In de laatste fase van de fitting worden ook beelden gebruikt om de kwaliteit van de parameterschatting te verbeteren.

Segmentatie is een zeer belangrijke stap die moet worden uitgevoerd als voorbereiding voor objectherkenning en modelfitting. We presenteren een methode voor segmentatie van puntenwolken die oversegmentatie voorkomt, terwijl de invoer wordt gepartitioneerd in niet-overlappende, gladde en samenhangende oppervlakken. Er wordt een criterium gebruikt dat gebaseerd is op een combinatie van overeenkomst in oppervlaktenormalen en ruimtelijke verbondenheid, wat we de gladheidsvoorwaarde noemen. Omdat we geen oppervlaktekromming gebruiken is ons algoritme minder gevoelig voor ruis. Verder zijn er maar een paar parameters die aangepast kunnen worden om een balans tussen onder- en oversegmentatie te vinden.

Segmentatie wordt gevolgd door een fase van objectherkenning die gebaseerd is op een vorm van Hough-transformatie voor automatische vlak- en cilinderdetectie in de puntenwolken. Voor vlakdetectie is de Hough-transformatie driedimensionaal. Voor de cilinderdetectie zou directe toepassing van de Hough-transformatie een 5D Hough-ruimte nodig hebben wat zeer onpraktisch is vanwege de hoeveelheid benodigd computergeheugen en rekencapaciteit. Om dit op te lossen presenteren we een benadering in twee stappen die een 2D en een 3D Hough-ruimte nodig heeft. In de eerste stap detecteren we sterke hypothesen voor de oriëntatie van de cilinder. De tweede stap schat de overige drie parameters van de cilinder, namelijk de straal en de positie.

Het fitten van vlakken, cilinders, bollen, tori en CSG-modellen is erg belangrijk voor datareductie. Voor het fitten van CSG-modellen presenteert dit proefschrift drie verschillende methoden voor het benaderen van de loodrechte afstand. Deze methoden worden vergeleken op basis van snelheid en precisie.

Ook presenteren we methoden voor de registratie van laserscans die gebruik maken van gemodelleerde objecten in de individuele scans. Omdat de beschikbare geometrische structuur wordt gebruikt, is het niet nodig om meetmerken te plaatsen. We presenteren hiervoor twee methoden, de indirecte en de directe genoemd. De indirecte methode is een snelle manier om benaderde waarden te verkrijgen, terwijl de directe methode vervolgens wordt gebruikt om de benaderde oplossing te verfijnen. Ook presenteren we technieken voor het automatisch vinden van corresponderende objecten voor het registreren van scans. De gepresenteerde technieken zijn gebaseerd op voorwaarde-voortplanting en gebruiken de geometrische informatie, die beschikbaar is uit eerder vastgestelde correspondenties, om zo de mogelijkheden voor nieuwe correspondenties uit te filteren.

Hoewel puntenwolken belangrijk zijn voor de automatisering vanwege hun expliciete 3D informatie, bieden beelden een aanvullende informatiebron omdat deze goed gedefinieerde randen van de objecten bevatten. We presenteren methoden voor fitten van CSG-modellen op een combinatie van puntenwolken en beelden. Ook presenteren we technieken voor de specificatie van geometrische voorwaarden tussen onderdelen van een CSG-boom en het gebruik van deze voorwaarden in het schattingsproces. Een classificatie van veel voorkomende geometrische voorwaarden en hun wiskundige formulering wordt eveneens gegeven.

We hopen dat de technieken, die in dit proefschrift gepresenteerd worden, zullen leiden tot een verbetering van de efficiëntie van de reconstructie en de kwaliteit van uit puntenwolken en beelden verkregen modellen van industriële installaties.

# Acknowledgements

Doing PhD was a long and exciting journey, which would not have been possible without the help and support of many people. First of all I am very thankful to my promoter Prof. George Vosselman who gave me the opportunity to pursue a PhD at TU Delft. Without his guidance, support and constant help the research work presented here would not have been possible.

I also owe many thanks to my daily supervisor Frank van den Heuvel whose help and support was always there for both technical and personal problems. His constructive criticism helped to focus my ideas and his constant encouragement kept me going in spite of many bumps in the road. I also owe him my gratitude for introducing me to many people working in similar fields.

My colleagues at the section of Photogrammetry and Remote Sensing (FRS) proved a great source of technical advice and helped to create an environment that was a joy to work in. I would especially like to thank Sander Dijkman, Daniel Rico, Norbert Pfeifer, Ben Gorte, Sagi Filin, Ildiko Suveg, Siyka Zlatanova, George Sithole, Fatemeh Karimi, Abbas Malian, Alexander Bucksch, Hieu Van Duong, and Roderik Lindenberg.

The other members of the section of FRS also deserve special thanks as they helped to solve many of the administrative problems and the journey would have been very difficult without their constant help and support. They are Magda Boeke, Lidwien de Jong-Bakker and Vera van Bragt. I am especially thankful to Lidwien for the help and support throught out the years.

Being away from my family was not easy, but many friends from Delft and other places provided another family away from home. I owe special thanks to all of my friends from the group of SSAHolland. In this regard Shabir Dar, Naveed, Khurrum Bokhari, Zeeshan, Haroon Awais, Nafees Khan, Shailaja and Raji deserve special mention.

I also owe many thanks to the friends from Holland Reading Club. Without the

social events arranged by them the sometimes heavy work of PhD would have been very difficult. I would like to especially mention Jacob, Martiza, Malika, Catarina, and Pablo.

I was fortunate enough to meet Rakhi Gupta during my stay in the Netherlands, whose companionship and support kept me going through many obstacles on the long path.

I would also like to thank Waltraut Stroh from International Students Chaplaincy for her motherly affection, support and guidance in all kind of problems.

I would also like to thanks my house mates at Breestraat 33, whose friendship created a home-like environment. They are Renier Ringers, Joel and Jorg. I would also thank Mr. and Mrs. van Bergen for their support and help through many problems.

Finally, I would like to thank my parents and siblings whose support and love was always there when I needed it most. I humbly dedicate this thesis to them.

Tahir Rabbani

# Contents

**List of Figures** **iv**

**List of Tables** **vii**

**1 Introduction** **1**
1.0.1 Motivation for using laser scanning . . . . . . . . . . . . . . . 3
1.0.2 The need for modeling . . . . . . . . . . . . . . . . . . . . . . 5
1.1 Literature survey . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
1.1.1 Segmentation . . . . . . . . . . . . . . . . . . . . . . . . . . 9
1.1.2 Surface Fitting . . . . . . . . . . . . . . . . . . . . . . . . . 12
1.1.3 Registration . . . . . . . . . . . . . . . . . . . . . . . . . . 16
1.1.4 Object recognition . . . . . . . . . . . . . . . . . . . . . . . 18
1.2 Objectives and modeling pipeline . . . . . . . . . . . . . . . . . . 22
1.3 Outline of the thesis . . . . . . . . . . . . . . . . . . . . . . . . . 24

**2 Segmentation using smoothness constraint** **27**
2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
2.1.1 Problem statement . . . . . . . . . . . . . . . . . . . . . . . 27
2.1.2 Previous work . . . . . . . . . . . . . . . . . . . . . . . . . . 28
2.1.3 Problems with existing methods . . . . . . . . . . . . . . . . . 29
2.1.4 Objectives and motivation . . . . . . . . . . . . . . . . . . . . 29
2.2 Segmentation algorithm . . . . . . . . . . . . . . . . . . . . . . . . 30
2.2.1 Normal estimation . . . . . . . . . . . . . . . . . . . . . . . . 31
2.2.2 Region growing . . . . . . . . . . . . . . . . . . . . . . . . . 34
2.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
2.4 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

**3 Object recognition** **41**
3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
3.2 Plane detection . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
3.3 Cylinder detection . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
3.3.1 Orientation Estimation . . . . . . . . . . . . . . . . . . . . . . 46

3.3.2 Position and Radius Estimation . . . . . . . . . . . . . . . . . 49
3.3.3 Example . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
3.3.4 Results of cylinder detection . . . . . . . . . . . . . . . . . . . 51
3.4 Ambiguity resolution by hypothesis testing . . . . . . . . . . . . . . . 53
3.4.1 Planarity test . . . . . . . . . . . . . . . . . . . . . . . . . 54
3.4.2 Cylindricality test . . . . . . . . . . . . . . . . . . . . . . . 55
3.5 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
3.6 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

**4 Model fitting to point clouds 59**
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
4.2 Problem definition . . . . . . . . . . . . . . . . . . . . . . . . . . 60
4.2.1 Fitting method . . . . . . . . . . . . . . . . . . . . . . . . . 61
4.2.2 Different distance measures . . . . . . . . . . . . . . . . . . . 61
4.3 Fitting of simple geometric objects . . . . . . . . . . . . . . . . . . 64
4.3.1 Quadric fitting for approximate values . . . . . . . . . . . . . 64
4.3.2 Plane fitting . . . . . . . . . . . . . . . . . . . . . . . . . . 66
4.3.3 Sphere fitting . . . . . . . . . . . . . . . . . . . . . . . . . . 67
4.3.4 Cylinder fitting . . . . . . . . . . . . . . . . . . . . . . . . . 68
4.3.5 Torus fitting . . . . . . . . . . . . . . . . . . . . . . . . . . 69
4.3.6 Cone fitting . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
4.4 CSG Fitting . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
4.4.1 Motivation for using CSG . . . . . . . . . . . . . . . . . . . . 71
4.4.2 Distance Approximation Algorithms . . . . . . . . . . . . . . . 73
4.4.3 Internal Constraints . . . . . . . . . . . . . . . . . . . . . . . 76
4.4.4 Fitting Algorithm . . . . . . . . . . . . . . . . . . . . . . . . 77
4.5 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
4.6 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79

**5 Model-based registration 81**
5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
5.1.1 Presented approach . . . . . . . . . . . . . . . . . . . . . . . 83
5.1.2 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
5.2 Notation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
5.3 Indirect Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
5.3.1 Spheres . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
5.3.2 Planes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
5.3.3 Cylinder . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
5.3.4 Torus . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
5.4 Direct Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
5.4.1 Plane . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
5.4.2 Sphere . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
5.4.3 Cylinder . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
5.4.4 Torus . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
5.5 Results and discussion . . . . . . . . . . . . . . . . . . . . . . . . 95
5.6 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101

**6 Automatic correspondence search 103**
6.1 Introduction . . . . . 104
6.1.1 Outline . . . . . 104
6.2 Algorithm outline . . . . . 104
6.2.1 Naming conventions . . . . . 104
6.2.2 Algorithm steps . . . . . 105
6.3 Constrained Search for Correspondences . . . . . 105
6.3.1 Planar matches . . . . . 106
6.3.2 Spherical matches . . . . . 109
6.3.3 Cylindrical matches . . . . . 111
6.3.4 Toroidal matches . . . . . 113
6.3.5 Mixed objects. . . . . . 114
6.4 Comparison . . . . . 114
6.5 Results . . . . . 116
6.6 Conclusions . . . . . 116

**7 Constrained CSG fitting 119**
7.1 Introduction . . . . . 120
7.2 Model Fitting . . . . . 122
7.2.1 Observation equations with constraints . . . . . 122
7.2.2 Fitting of a CSG model . . . . . 124
7.3 Types of constraints . . . . . 125
7.3.1 Parameter constraint . . . . . 126
7.3.2 Rotation constraint . . . . . 126
7.3.3 Translation constraint . . . . . 127
7.3.4 Connection constraint . . . . . 128
7.3.5 Bounded translation constraint . . . . . 128
7.4 Examples of constraints . . . . . 129
7.4.1 Flanged T-Junction . . . . . 129
7.4.2 Flanged curved pipe . . . . . 129
7.5 Fitting Experiments . . . . . 130
7.5.1 Cylinder fitting . . . . . 131
7.5.2 Box Fitting . . . . . 132
7.6 Results . . . . . 133
7.7 Conclusions . . . . . 134

**8 Conclusions 135**
8.1 Achievements . . . . . 135
8.2 Directions for future work . . . . . 137

**A Uniform sampling of the orientation space 139**

**Bibliography 153**

# List of Figures

1.1 New generation of laser scanners . . . . . . . . . . . . . . . . . . . . . 5
1.2 Comparison of point clouds, images and CAD model . . . . . . . . . 6
1.3 Flowchart of our modeling pipeline . . . . . . . . . . . . . . . . . . . . 22

2.1 Flowchart of segmentation method . . . . . . . . . . . . . . . . . . . 30
2.2 Plane fitting to the neighborhood of a point . . . . . . . . . . . . . . 33
2.3 Comparison of segmentation for a toroidal surface . . . . . . . . . . 36
2.4 Results of segmentation . . . . . . . . . . . . . . . . . . . . . . . . . . 37
2.5 Effects of $r_{th}$ on segmentation . . . . . . . . . . . . . . . . . . . . . 38

3.1 Voting in the Hough space for plane detection . . . . . . . . . . . . . 43
3.2 Discretized Hough space for the points in a plane . . . . . . . . . . . 44
3.3 The five parameters of a cylinder . . . . . . . . . . . . . . . . . . . . 45
3.4 Each cylinder results in a great circle on the Gaussian sphere . . . . . 46
3.5 Voting scheme for one point on the input Gaussian sphere . . . . . . 47
3.6 Step 1: orientation estimation . . . . . . . . . . . . . . . . . . . . . . 48
3.7 Step 2: position and radius estimation . . . . . . . . . . . . . . . . . 49
3.8 Step-by-step processing of the point cloud of an L-junction . . . . . . 52
3.9 Results of cylinder detection on industrial point clouds . . . . . . . . 53
3.10 Ambiguity between the cylinder and the plane . . . . . . . . . . . . . 53
3.11 Planarity test . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
3.12 Cylindricality test . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
3.13 Results of simultaneous plane and cylinder detection . . . . . . . . . 57

4.1 The algebraic vs the geometric distance fitting for a sphere . . . . . . 63
4.2 Parameters of the object models . . . . . . . . . . . . . . . . . . . . . 67
4.3 Approximate values through quadric fitting . . . . . . . . . . . . . . 68
4.4 Comparison of CSG and B-rep fitting . . . . . . . . . . . . . . . . . . 71
4.5 Constructive solid geometry (CSG) for industrial models . . . . . . . 71
4.6 Three approximation methods for CSG distance calculation . . . . . 73
4.7 Errors in ICP$^2$ and ICT . . . . . . . . . . . . . . . . . . . . . . . . . 75

4.8 Importance of constraints for CSG fitting . . . . . . . . . . . . . . . . . 75
4.9 The reduction in the parameters by using constraints . . . . . . . . . 76
4.10 Results of CSG fitting . . . . . . . . . . . . . . . . . . . . . . . . . . 78

5.1 Objects used as targets for registration . . . . . . . . . . . . . . . . . 83
5.2 Flowchart of the Direct and Indirect method . . . . . . . . . . . . . . 84
5.3 Parameters of the objects used as targets . . . . . . . . . . . . . . . . 88
5.4 Scans used for registration . . . . . . . . . . . . . . . . . . . . . . . . 95
5.5 Results of Indirect method . . . . . . . . . . . . . . . . . . . . . . . . 96
5.6 Results of combining objects after Indirect method . . . . . . . . . . 97
5.7 Improvements obtained through Direct method . . . . . . . . . . . . 98
5.8 ICP vs Direct method . . . . . . . . . . . . . . . . . . . . . . . . . . . 99
5.9 Cylinders used for comparing ICP with Direct method . . . . . . . . . 100

6.1 Constrained correspondence search for planes . . . . . . . . . . . . . 106
6.2 Constrained correspondence search for sphere . . . . . . . . . . . . . 110
6.3 Constrained correspondence search for cylinders . . . . . . . . . . . 112
6.4 Constrained correspondence search for torus . . . . . . . . . . . . . . 113
6.5 Results of processing an industrial data-set . . . . . . . . . . . . . . . 117
6.6 The improvements obtained by integrated adjustment . . . . . . . . 117

7.1 CSG fitting – point cloud vs image . . . . . . . . . . . . . . . . . . . 124
7.2 Constraints for two CSG objects . . . . . . . . . . . . . . . . . . . . . 125
7.3 Connection constraint . . . . . . . . . . . . . . . . . . . . . . . . . . 127
7.4 Bounded translation constraint . . . . . . . . . . . . . . . . . . . . . 128
7.5 Experiment for cylinder fitting . . . . . . . . . . . . . . . . . . . . . . 131
7.6 Experiment for box fitting . . . . . . . . . . . . . . . . . . . . . . . . 132
7.7 Catalog of CSG objects . . . . . . . . . . . . . . . . . . . . . . . . . . 133
7.8 Modeling of an industrial installation . . . . . . . . . . . . . . . . . . 134

A.1 Different methods of sampling a unit sphere . . . . . . . . . . . . . . 139

# List of Tables

1.1 Comparison of different modeling techniques . . . . . . . . . . . . . 7

4.1 Comparison of fitting based on the algebraic and the geometric distance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
4.2 Quadric types used for approximation . . . . . . . . . . . . . . . . . 65
4.3 Comparison of ICS, ICT and ICP$^2$ . . . . . . . . . . . . . . . . . . 80

5.1 Registration parameters from Indirect method . . . . . . . . . . . . . 97
5.2 Registration parameters from Direct method . . . . . . . . . . . . . . 97
5.3 Standard deviations from Direct method . . . . . . . . . . . . . . . . 98
5.4 Comparison of ICP and Direct method. Parameter values and average residuals . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100
5.5 Comparison of ICP and Direct method. Parameter standard deviations.100

6.1 Comparison of different search methods . . . . . . . . . . . . . . . . 116

7.1 Standard deviations for Cylinder fitting experiment . . . . . . . . . . 131
7.2 Standard deviations for Box fitting experiment . . . . . . . . . . . . 132

# Chapter 1

# Introduction

> In science men have discovered an activity of the very highest value in which they are no longer, as in art, dependent for progress upon the appearance of continually greater genius, for in science the successors stand upon the shoulders of their predecessors.
>
> Bertrand Russell (1872 - 1970)

> A dwarf sees farther than the giant when he has the giant's shoulders to mount on.
>
> S. T. Coleridge

The objective of this thesis is to present new methods and techniques which can be used for automatic or more efficient semiautomatic 3D modeling of existing industrial installations from point clouds and images. The goal is to use explicit 3D information from the point clouds to automatically detect the objects and structure present in the scene. The detected objects are then used as targets for model based registration, which can be automated by searching for object correspondences. To avoid manual editing the presented techniques use models from a catalog of commonly found CAD objects as templates for model fitting. In the final fitting phase images are also included to improve the quality of parameter estimation. To lay the ground work in this chapter we will review the application areas of industrial reconstruction and present a survey of the previous work. Finally, we will give an outline of the thesis.

There is an increasing demand for accurate, as-built, 3D models of existing industrial sites in many sectors. The following are some of the application areas which either require or can benefit from the availability of such models:

- Planning (clash detection, decommissioning, design changes)
- Revamping and retrofitting of old sites
- Implementation of services based on Virtual and Augmented reality.

- Off-site training
- Safety analysis
- Change detection.

Although most of the new industrial sites are designed using 3D CAD techniques, in most cases the initial model represents a functional design rather than the final as-built situation. Moreover, industrial facilities are often very dynamic environments, where constant changes are required to improve health and safety, to increase efficiency, and to reduce hazardous emissions in accordance with the environmental regulations. As a result after a few years there is a big gap between the documented model and the as-built situation. In most cases it is not cost-effective or practical to update these models at the end of construction or after each and every change. For old and legacy sites the situation is even worse, as most of them were initially designed using old 2D CAD techniques and there is no 3D model available. Consequently, in both cases, when new changes are planned up-to-date as-built 3D information is required.

Traditional methods for acquiring as-built information consist of manual measurements by tape and tacheometry. As-built modeling using measuring-tape is accurate only up to 25–75 mm (Sanders, 2001) which is not acceptable for most planning scenarios where measurement accuracy of ± 2mm is usually required. Tacheometry provides high accuracy, but due to its slow speed combined with the limitations of the measurement technique, the density of the measured points is very low. As a result, the sparse 3D measurements have to be manually extrapolated to make an approximate 3D model which, except on the points explicitly measured, does not provide a true and accurate picture of the as-built situation. Moreover, as most industrial sites contain many curved faces like cylinders and bends, the sparse point clouds from tacheometry become even more inadequate.

For the cost-effective management of complex industrial facilities such as nuclear power plants, offshore oil production platforms, and petrochemical refineries it is imperative that they constantly operate near their peak productivity level. As profits are affected by down time, each and every change and refurbishment activity should be carefully planned to reduce costs arising from unanticipated delays. Without detailed planning the projects cannot be completed within agreed schedules and budgets. The retrofits planned using inaccurate as-built models will result in higher shutdown time, and ultimately higher costs. Moreover, due to the continuous inflow of material there are maximum limits on the available shut down time, which cannot be violated without incurring extra costs. For example in the case of the retrofit project of a Chevron plant the shut-down time was limited to 72 hours (Sanders, 2001).

Many industrial facilities provide a limited personnel access because of the presence of radioactive, toxic or hazardous materials together with an unsafe working environment (Mensi, 2002a; Pot et al., 1997). In such situations, the use of traditional contact-based measurement methods is not possible and non-contact measurement methods have to be used.

Different solutions based on traditional photogrammetry have been proposed in literature, which provide rapid, accurate, and relatively dense 3D measurements. Photogrammetry provides a set of techniques for accurate 2D and 3D measurements from photographs (Atkinson, 1996). Some photogrammetric solutions currently available for industrial reconstruction are Shape capture & Shape quest (`www.shapecapture.com`), Photomodeler (`www.photomodeler.com/`), RolleiMetric CDW (`www.rollei.de`), Invers PHAUST(`www.invers-essen.de`), and Hazmap (`www.absl.co.uk`) (Chapman et al., 2004).

Traditional photogrammetry depends on point or line measurements from which it is very hard to get complete CAD models without extensive manual editing and refinement. To address this problem CAD based photogrammetric techniques have been developed at the section of Photogrammetry and Remote sensing at TU Delft. This has resulted in a software package called *Piper* (Ermes et al., 1999; Tangelder et al., 1999, 2003). Piper allows the user to select a model from a catalog of CAD model templates, drop it on the set of oriented images, and by aligning the back-projected contours of the model to the edges visible in the images, the shape and pose parameters of the model can be calculated. Moreover, various geometrical and topological constraints for the model can be specified, which decrease the degrees of freedom resulting in a better estimation of parameters with less manual work. Using the user-supplied constraints and initial pose as a starting point, the software calculates the best-fit solution using non-linear least squares techniques. In spite of these improvements, the manual manipulation of the selected model to give a good initial alignment can be quite labor-intensive. The complexity of the industrial environments, which results in a high degree of occlusions and clutter, makes the situation even worse. Furthermore, as photographs do not contain any explicit 3D information, the prospects of automation are very dim.

Compared to photogrammetry laser scanning provides explicit and dense 3D measurements. There has been a rapid increase in the speed and accuracy of the laser scanners in the last decade, while their costs and sizes has been continuously shrinking (Blais, 2004; Laser scanner survey, 2005). As a result, the use of laser scanning technologies for 3D data capture for industrial reconstruction has grown considerably over the last few years (Sanins, 2004).

### 1.0.1 Motivation for using laser scanning

Some of the reasons leading to a rapid acceptance and usage of laser scanning for industrial reconstruction are as follows:

**Explicit 3D information** Laser scanning is an active measurement technique (Jahne et al., 1999, chap. 7), which makes it 'almost' independent of the lighting conditions and surface texture. In contrast photogrammetry requires either the presence of good surface texture or its introduction through placement of stickers and targets or by using structured light. Even for CAD based photogrammetric techniques which fit image gradients to the back-projected model contours good contrast on the image-edges is required(Ermes et al., 1999; Tangelder et al., 1999). Due to the complexity of the industrial sites the explicit 3D information available from the point clouds is very attractive because this enables the rapid and accurate capture of the geometry of a complex scene.

**High density** As mentioned before the density of 3d measurements from traditional survey methods and photogrammetry is very low, which can be converted to a CAD model only after extensive manual editing and extrapolation. In contrast the current generation of laser scanners provide high density 3D measurements. One point every 5mm is quite typical (Blais, 2004; Laser scanner survey, 2005). This high density provides enough raw data from which accurate and detailed 3D models can be obtained.

**Non-contact measurement** Similar to photogrammetry, laser scanning is a non-contact measurement method. Typical working distance of laser scanners is 50-200m, which makes them suitable for inaccessible and hostile environments. Additionally, as most scanner use eye-safe lasers, the scanning can be done without disturbing the usual operation of the industrial facility, which reduces the down-time required for data capture.

**High speed** The measurement speed of laser scanners has been increasing continuously during last few years. A typical scan takes 10-15 minutes, while providing a few million 3D points.

**Large FOV** Although the previous generation of laser scanners provided a limited field of view (FOV), panoramic scanning is rapidly becoming a norm (iQvolution, 2005; Laser scanner survey, 2005; Mensi, 2005; Riegl, 2005). The large FOV reduces overall acquisition time by decreasing the manual shifting of the equipment. Additionally, it provides better registration by ensuring high overlap and stronger geometry.

**Varying level of detail** Most laser scanners provide scanning abilities with different levels of detail. This enables scanning of the interesting parts with high point density, while still keeping the surrounding information for global registration.

**Integrated imagery** Most scanners of the current generation are coming with inbuilt high resolution cameras, which provide registered images. For example Mensi GS200 (Figure 1.1(a)) provides a 9 mega pixel image with live video (Mensi, 2005). Similarly Riegl LMS-360i (Figure 1.1(b)) provides a registered panoramic image(Riegl, 2005). Other scanners are coming with similar facilities. The registered imagery makes possible the simultaneous use of measurement techniques from photogrammetry. Moreover, images provide color and texture information which can be useful for human interpretation and photo-realistic visualization.

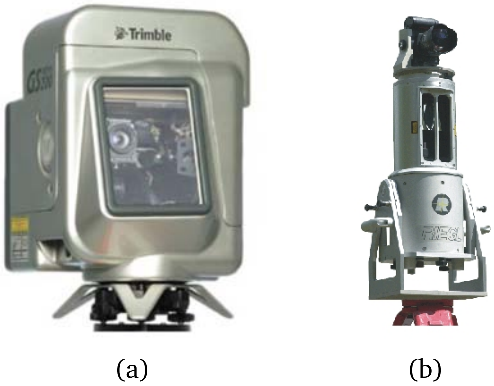

(a) (b)

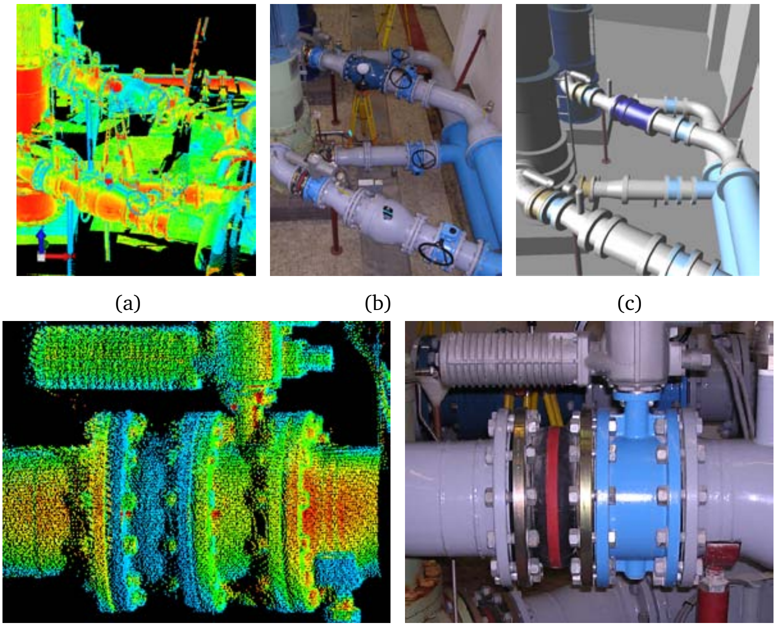
**Figure 1.1:** *New generation of laser scanners provide better speed, high accuracy, big FOV, and come with inbuilt image acquisition capabilities (a) Mensi GS200 (source Mensi, 2005)(b) Riegl LMS-360i (source Riegl, 2005)*

### 1.0.2 The need for modeling

The 3D measurements from the laser scanner come in the form of a point cloud. Due to occlusions and a limited FOV of the scanner, the whole installation cannot be covered from one location, and data has to be acquired from multiple different viewpoints. To bring different scans in a consistent coordinate system, they are registered by calculating inter-scan rigid body transformations. After registration we have one point cloud representing the complete measured data in one consistent coordinate system.

For converting the point cloud to a CAD representation we need the step of modeling, where different types of surfaces are fitted to a selected subset of the point cloud. The resulting surfaces are edited, extended, and intersected to get a full 3D model. Modeling is one of the most time consuming and costly processes during the reconstruction of any industrial installation. This cost arises due to the high amount of manual input required from the human operator. Although the situation has improved a lot compared to the approaches based on traditional photogrammetry, still most of the current point cloud processing softwares provide minimal if any automatic modeling facilities (Böhler et al., 2002).

A question is often raised about the need for modeling. If it is fast and cheap to acquire dense point clouds using the current generation of laser scanners, why not use the point cloud directly instead of deriving a model from it? It is argued that a dense point cloud should be as good as a CAD model. There are certain weaknesses with this argument. In any typical industrial site it is practically impossible to get a complete coverage of all areas by point clouds. Even if issues of time and effort involved are ignored, a high degree of occlusions arising from the clutter and complexity often makes placement of the scanner at arbitrary positions impossible. The resultant incomplete point cloud, although good for simple visualization, cannot be directly employed for tasks like planning and clash detection, which require a complete as-built picture of the environment. For example compare the point cloud of a typical industrial site in Figure 1.2(a) with its 3D model in 1.2(c). There are many holes in the point cloud data due to occlusions and most of the pipes are not completely covered. However, by fitting models the missing data



(a) (b) (c)

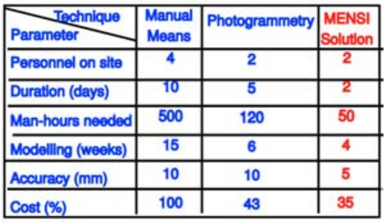
(d) (e)

**Figure 1.2:** *A comparison of point cloud, image and CAD model (a) Point cloud provides explicit 3D information but contains incomplete information (b) images provide color and texture (c) by fitting models to point clouds a CAD description can be obtained that provides directly usable, complete, and compact 3D information of as-built situation (d-e) point clouds are generally of low quality near the object edges where images provide better information (Point cloud and model provided by (Fugro, 2006))*

has been estimated along with a high degree of compression in the amount of information from millions of points to a few parameters for each object in the fitted model.

The engineering work flow in most industries is based on working with standard 2D and 3D CAD models which means that introduction of the point cloud as a new geometry representation has inherent integration problems. The space and time complexity of manipulating, storing and sharing huge amounts of data produced by laser scanners adds another dimension to this problem.

Although the quality of point cloud data has improved a lot due to the development of better measurement and processing techniques of the past decade, it is still worse by a factor of 5 or more compared to traditional surveying instruments like total station. For example whereas the accuracy of Leica HDS2500 laser scanner is 5mm at 100m (Leica HDS, 2005), for Leica TC2003 total station the accuracy is 1 mm over the range of 2.5 to 3.5 km (Leica TPS, 2005). However, the effects of noise in the measured point cloud can be averaged by fitting models. When a model is fitted to thousands of points, the errors in the estimated parameters are reduced



| Technique / Parameter | Manual Means | Photogrammetry | MENSI Solution |
|---|---|---|---|
| Personnel on site | 4 | 2 | 2 |
| Duration (days) | 10 | 5 | 2 |
| Man-hours needed | 500 | 120 | 50 |
| Modelling (weeks) | 15 | 6 | 4 |
| Accuracy (mm) | 10 | 10 | 5 |
| Cost (%) | 100 | 43 | 35 |

**Table 1.1:** *Comparison of different techniques for as-built modeling of a Shell off-shore plant. (source Mensi, 2002b)*

significantly. Additionally the stage of fitting, if based on least squares, provides measures about the quality of the estimated parameters which can prove valuable for decision making.

To summarize, modeling is necessary because it provides a complete picture by filling in the gaps left by occlusions, averages the effects of noise providing better accuracy, reduces the amount of data and the resulting CAD models fits nicely in the engineering work flow of daily industrial practice.

Almost all modeling tools available on the market depend on heavy operator intervention for most of the modeling tasks. Although there are some semi-automatic tools like plane or cylinder growing but even there the operator has to start the growing process for each primitive. Furthermore, the fitted surfaces must be manually edited by the operator to convert them to a CAD description.

To look at some example scenarios reported in literature, during the revamp of a Chevron installation, which was done using laser scanning, 40% of the costs came from data-processing labor (Sanders, 2001). In Table 1.1 , reproduced from Mensi (2002b), the time estimates for as-built modeling of an off-shore of Shell using different modeling techniques are given. Compared to photogrammetry there is a substantial improvement for laser scanning in terms of time and labor required. However, modeling still remains the major bottleneck. The estimated time for data acquisition using laser scanning was 2 days, but the modeling was estimated to take 4 weeks, which is more by a factor of 10.

Although the processes required for 3D reconstruction from point clouds (range data) and images have extensive literature available, the problem as applied to the automation of modeling, especially for industrial reconstruction, is far from solved, and needs further research efforts. The presence of well-defined CAD objects on industrial sites, along with their inherent geometric and topological constraints, provides rich source of a priori information which can be used for the development of automatic or semiautomatic solutions for industrial reconstruction.

The rest of the chapter is structured as follows. In Section 1.1 we present a literature survey of the different reported methods and techniques for processing range data and 2D images for 3D reconstruction. This includes a discussion of the problems of segmentation, model fitting, registration and object recognition. Section 1.2 contains our framework for the reconstruction of industrial facilities. An outline of the thesis is presented in Section 1.3.

## 1.1 Literature survey

As noted before, the commercial laser scanners available today are capable of providing highly accurate, dense point clouds of big installations at high speeds. They can be employed in daylight or at night and the eye-safe lasers further facilitate their use. The state of the art of point cloud data processing lags far behind that of the data acquisition. The situation is further complicated by often conflicting requirements of different user-communities that work with the range data. This makes the emergence of commonly useful techniques quite unlikely.

Computer graphics, computer vision and reverse engineering are three important fields that have been using range data for different applications. Traditional close-range photogrammetry has concentrated more on using 2D images and has started using the range data only recently. While computer graphics community strives for visual realism, computer vision needs the models as an input for various automation processes like vehicle guidance and robot navigation; there high speed is the main requirement, even if it is at the cost of data quality. The users from reverse engineering community require the final model to be a high fidelity representation of the true object geometry. In contrast, the computer graphics community gives more stress to visual realism at low computational cost.

The types of surfaces which are modeled in each field also vary. Computer graphics aims for free form or sculptured surfaces like human models for games; computer vision focuses on modeling the inside geometry of buildings which mostly consist of planar faces, and reverse engineering has to work with B-splines or NURBS as they are commonly used CAD primitives for mechanical objects.

These differences in requirements have led to the development of very application-specific data processing methods. Due to the conflicting trade-offs involved, the techniques developed by one user community cannot be directly used by another. However, the basic ideas are still similar and thus can be mutually useful.

It is worthwhile to survey the research efforts that have been put into various range data processing techniques, though it cannot be expected that the reported methods can be directly applied for the task of industrial reconstruction. Reconstruction of industrial installations is expected to have many similarities to reverse engineering because the final model must be similar to a CAD design and must have high fidelity. However, the CAD primitives for industrial installations are quite different from the ones used for reverse-engineered parts. Furthermore, the constraints between different primitives in industrial facilities can be more easily specified, and thus should prove more useful during modeling.

In the following sections we survey the literature related to the following processes:

I. Segmentation

II. Surface fitting

III. Registration

IV. Object recognition

### 1.1.1 Segmentation

Segmentation is the process of dividing a given point cloud or range image into a number of disjoint subsets each of which is spatially connected. Ideally each surface should result in a separate segment. If a surface is divided into more than one segments, it is called *over-segmentation*. Similarly, if more than one surfaces are combined together in one segment the resulting situation is called *under-segmentation*. The problem of segmentation is quite similar to that of clustering in pattern recognition which tries to partition a given dataset into mutually disjoint groups such that a chosen criterion is optimized (Fukunaga, 1990; Webb, 1999). Clustering usually works in a feature space rather than working on the raw data. Due to this similarity many segmentation methods have borrowed or adapted ideas from clustering.

Various approaches for segmentation differ mainly in the method or criterion used to measure the similarity between a given set of points and hence for making the grouping decisions. Once such a similarity measure has been defined, segments can be obtained by grouping together the points whose similarity measure is within given thresholds and which are spatially connected. Most of the segmentation methods presented in the literature are for depth-maps as due to their $2\frac{1}{2}$D nature operations from traditional image processing can be directly applied.

There are mainly three varieties of range segmentation algorithms:

(a) Edge-based segmentation

(b) Surface-based segmentation

(c) Scan-line based segmentation

#### Edge-based segmentation

Edge based segmentation algorithms have two main stages: edge detection which outlines the borders of different regions, followed by the grouping of the points inside the boundaries giving the final segments. Edges in a given depth map are defined by the points where changes in the local surface properties exceed a given threshold. The local surface properties mostly used are surface normals, gradients, principal curvatures, or higher order derivatives. As edge detection methods look for abrupt changes, they are very sensitive to the noise in the range data. Moreover, as only the measurements near the edges are used to make major decisions, the available information is not optimally utilized. In many situations the edges do not form closed boundary curves and it can be difficult to make correct grouping decisions resulting in over- or under-segmentation. Some of the typical variations on the edge-based segmentation techniques are reported by Bellon et al. (1999);

Bhanu et al. (1986); Jiang et al. (1995); Sappa and Devy (2001); Wani and Arabnia (2003); Wani and Batchelor (1994).

### Surface-based segmentation

The surface based segmentation methods use local surface properties as a similarity measure and merge together the points which are spatially close and have similar surface properties. These methods are relatively less sensitive to the noise in the data, and usually perform better when compared to edge based methods (Várady et al., 1997). The measure of similarity is usually based on local surface properties. For surface-based segmentation methods each point is directly assigned to one segment; unlike edge based methods there is no need to identify the surface boundaries.

One common problem faced by all range segmentation algorithms is the estimation of local surface properties like gradient, surface normal, principal curvatures and higher-order derivatives from the noisy data. Robust estimation techniques must be used to counter the effects of outliers and noise (Flynn and Jain, 1988; Stewart, 1999; Tang and Medioni, 1999).

For surface based segmentation methods two approaches are possible: bottom-up and top-down. Bottom up approaches start from some seed-pixels and grow the segments from there based on the given similarity criterion. The selection of the seed points is important because the final segmentation results are dependent on it. Top-down methods start by assigning all the pixels to one group and fitting a single surface to it. Then as long as a chosen figure of merit for fitting is higher than a threshold they keep on subdividing the region (Parvin and Medioni, 1986; Xiang and Wang, 2004). Most of the reported methods for range segmentation use bottom-up strategy.

It is important to have a sufficient number of different surface classes available during the segmentation process, because failing to do so can lead to either over-segmentation or under-segmentation. For example if only planar patches are used and the scene contains curved objects like cylinders or spheres, the range data would be highly over-segmented.

The inclusion of complex surfaces as possible candidates is accompanied with a caveat. The estimation of the parameters describing a complex surface from the noisy range data is usually poor, and so is the resulting segmentation. The situation here is similar to what is referred to as ‘curse of dimensionality’ in pattern recognition (Bellman, 1961; Bishop, 1996). As the number of allowed shapes and their complexity increases, the estimation of the required parameters from the noisy data becomes more and more difficult.

Although segmentation techniques incorporating general quadrics (Besl and Jain, 1988a) and super-quadrics (Gupta and Bajcsy, 1993)) have been developed, the results on actual data are not very satisfactory. As observed by Várady et al. (1997) better results can be obtained by considering only natural quadrics, as they need small number of parameters which have geometrical rather than algebraic significance. A method for faithful recovery of quadrics from range data is presented by Werghi et al. (1999a). A survey of recovering quadric surfaces in triangulated meshes is given by Petitjean (2002).

A superquadric defines a closed surface, whose parameters can be varied to change the resulting shape from a cube to a diamond to a pyramid and many intermediate forms. It is also possible to specify the scale along each axis, bending along two axes and tapering along one of the axis. This gives a compact representation that can cover the shape of many geometrical primitives. However, the fitting of superquadrics requires the estimation of many non-linear parameters from the range data, which becomes problematic in the presence of noise. Still assuming dense range data with low noise, their extraction has been used by many researchers. For example see Krivic and Solina (2004); Leonardis et al. (1997); Zhang (2003).

### Scanline-based segmentation

The third category of range segmentation methods is based on scan-line grouping. In the case of range images each row can be considered a scan-line, which can be treated independently of other scan-lines in the first stage. A scan-line grouping based segmentation method is presented by Jiang et al. (1996) for the extraction of planar segments from the range image. It uses the fact that a scan line on any 3D plane makes a 3D line. It detects the line segments in the first stage, followed by the grouping of the adjacent lines with similar properties to form planar segments. Some typical variations on this method are presented by Natonek (1998) and Khalifa et al. (2003). As in a point clouds there are no inherent scan-lines, Sithole and Vosselman (2003) have used profiles in different directions for the segmentation of air-borne laser scanner data. These profiles are generated by collecting points within a cylindrical volume around a given direction.

### Other approaches

A hybrid approach combining both edge cues and surface information for range segmentation is presented by Zhao and Zhang (1997). Edge and critical point detection, triangulation, and region growing are three main steps of the reported technique. Good segmentation of the range images, consisting of only polyhedral objects, are reported.

A range segmentation algorithm based on graph partitioning using normalized cut framework has been presented by Yu et al. (2001). The normalized cut criterion was defined by Shi and Malik (1997) for image segmentation; it measures both the total dissimilarity between the different groups as well as the total similarity within the groups. Yu et al. (2001) first detects connected clusters in the range image, and then uses average position, surface normal and average intensity of these clusters to group them together using normalized cuts. This algorithm results in over-segmentation which needs to be corrected by manual editing and merging.

Hoover et al. (1995) presented a methodology for the automatic comparison of different range segmentation algorithms, based on which four range segmentation methods for planar surfaces were compared by Hoover et al. (1996). Powell et al. (1998) extends this comparison to two methods for segmentation of the curved objects. The methods compared were by Besl and Jain (1988b) and Jiang et al. (1996).

### Point cloud vs range image

The methods tested by Hoover et al. (1996) and by Powell et al. (1998) assume that the input consists of range images sampled on a regular two-dimensional grid, resulting in $2\frac{1}{2}$D data. This requirement, although satisfied by most structured-light based scanners, does not apply to most of the laser range scanners which provide data in the form of an unstructured point clouds. There are two solutions to this problem:

1. The point cloud can be resampled to convert it to the data defined on a regular 2D or 3D grid. This resampling can use either splatting or interpolation. Splatting (Schroeder et al., 1998) converts an unstructured point cloud to a structured grid by replacing each point by an influence function which is usually a Gaussian. Moreover, the splats can be made to change shape in accordance with the local surface properties, leading to a more faithful surface representation after resampling. Alternatively, resampling based on linear, quadratic, or cubic interpolation of the point cloud can be used.
2. The second option is to build a topology directly from the point cloud by using triangulation. A triangulation consists of n-dimensional simplices that completely bound the points and their convex hull (Goodman and O'Rourke, 1997). The result of triangulation is a set of triangles in case of 2D and $2\frac{1}{2}$D data and a set of tetrahedra in the case of 3D data. Delaunay triangulation is a special case requiring the circumsphere of each simplex in the final result to contain no points other than the ones defining it (O'Rourke, 1998). For 2D data Delaunay triangulation has been shown to be the optimal triangulation in terms of producing the triangles with the best interior angles (by maximizing the minimum angle). Furthermore, the Delaunay triangulation and Voronoi diagram in $\mathbb{R}^2$ are dual to each other.

For processing unstructured 3D data, without performing any of the above mentioned conversions, it is necessary to build neighborhood relationships to estimate local surface properties. The approach to select k-nearest neighbors is computationally expensive but can be optimized by using space-partitioning methods like k-d trees for efficient localized searches (Arya et al., 1998; Goodman and O'Rourke, 1997).

### 1.1.2 Surface Fitting

The problem of fitting CAD models to point clouds arises in many applications like model-based object recognition, surface reconstruction, reverse engineering and quality control. Recent advances in laser scanning technologies have also added to their importance, as acquisition of dense point clouds has become both faster and cost-effective.

As pointed by Várady et al. (1997) segmentation and surface fitting can be regarded as a 'chicken and egg' problem, because if a priori information about the surfaces and their location is available we can just pick the points which are within a small distance of the surface. Similarly, if we know that a certain group of pixels belong to one surface, we can easily find the type of the surface they represent.

There are different methods of representing surfaces for fitting. The choice of representation is important as it influences the complexity of the resulting method and can limit the types of models that can be faithfully fitted. The following are some of the available choices for representing surfaces (Foley et al., 1990):

**Explicit function** For 3D surfaces this representation expresses one of the coordinates as a function of the remaining two coordinates and their bounds. Mathematically it can be given as:

$$z = f(x, y) \qquad x_1 \leq x \leq x_2 \qquad y_1 \leq y \leq y_2 \tag{1.1}$$

Explicit functions can be used only for those surfaces which have a single $z$ value for each $(x, y)$ pair. This condition is not satisfied by most of the common objects found on industrial facilities. For example, a cylinder or a sphere cannot be represented in this form.

Explicit function can be used for plane fitting using the following equation:

$$z = ax + by + c \tag{1.2}$$

The expression in Equation 1.2 can faithfully represent only non-vertical planes. Moreover, least squares fitting based on this formulation minimizes only the differences in $z$ instead of the orthogonal distance of the point from the plane which is justified if only the measurement of $z$-coordinate contains noise, and $x$ and $y$ are without measurement errors (Press et al., 1988, chap. 15.3). As this assumption is not valid, the fitting based on the explicit function formulation, even when possible, results in poor estimation.

**Implicit function** In this representation the surface is defined as the zero-set of a three-dimensional function of $x$, $y$ and $z$.

$$f(x, y, z) = 0 \tag{1.3}$$

For example using this representation a sphere is defined as follows:

$$(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - r^2 = 0 \tag{1.4}$$

Where $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}$ is the center of the sphere and $r$ is its radius.

Similar implicit functions exist for planes (linear), cylinders (quadric), cones (quadric) and tori (quartic).

**Parametric surface** A parametric representation consists of a separate function for each of the $x$, $y$, and $z$ coordinates. Each of these functions is parametrized in terms of two new variables $u$ and $v$ (hence the name parametric). The domains of $u$ and $v$ are also given.

$$x = f(u, v) \qquad y = g(u, v) \qquad z = h(u, v) \tag{1.5}$$

For example the parametric representation of the sphere is as follows:

$$x = r\cos\theta\sin\phi + c_x \qquad y = r\sin\theta\sin\varphi + c_y \qquad z = r\cos\phi + c_z \tag{1.6}$$

Where $\theta$ with $0 \leq \theta < 2\pi$ is the azimuthal coordinate , $\varphi$ with $0 \leq \varphi < \pi$ is the polar coordinate

B-splines and NURBS are two of the most commonly used parametric surfaces in CAD applications. B-splines are a generalization of the Bézier curve, and use control points and basis function to represent smooth surface patches (Farin, 2002). NURBS are an extension of B-Splines and can also represent conics faithfully. The fitting procedure for both of them are highly non-linear. As the automatic selection of knots and control points is very difficult, usually it has to be done manually. B-splines and NURBS are necessary for reverse engineering of mechanical or other industrial parts, but do not find much use in the design of industrial facilities. Some typical techniques for their fitting to point clouds, often requiring manual specification of knots and good initial values, are reported by Forsey and Bartels (1995); Ma and Kruth (1995); Pottmann et al. (2002); Wang et al. (2004).

A survey of surface fitting methods is given by Söderkvist (1999).

### Choice of distance measures

All methods of surface fitting require a criterion that gives the degree of agreement or disagreement between the estimated surface and the measured point cloud. A distance measure is usually employed for this purpose. Two commonly used distance measures for fitting surfaces to point clouds are the algebraic and the orthogonal or geometric distances.

The algebraic distance is defined only for those surfaces which can be expressed as an implicit function. Surface fitting based on the algebraic distance can be expressed as a linear least squares problem, which has a closed form solution and can be readily solved by linear equation solvers. In contrast, for orthogonal distance fitting it is necessary to use iterative non-linear least squares techniques. Taubin (1993) has proposed a first order approximation from algebraic to the orthogonal distance that can be used to avoid iterative procedures, but the approximation is reported to be biased and results produced are usually inferior to those from the orthogonal distance fitting. A comparison of algebraic, orthogonal, and approximate-orthogonal distance for conic fitting to 2D data is given by Faber and Fisher (2001a,b).

Orthogonal distance of a point to a given surface is defined by its distance to the closest point on the surface. It is sometimes also called geometric distance. It provides a better measure of similarity compared to the algebraic distance. However, the fitting methods based on the orthogonal distance must use a non-linear least squares estimation procedure. Helfrich and Zwick (1993) and Ahn et al. (2002) give two different methods for the fitting of implicit surfaces using the orthogonal distance.

There has been a lot of work, mainly in the Computer Graphics community, on using the zero set of the signed distance field for the recovery of a smooth manifold from a given point cloud (Curless and Levoy, 1996; Hoppe et al., 1992; Masuda, 2002). Such a manifold is essentially a free-form surface, and thus cannot be easily represented by a CAD model. Moreover, the surface should be completely covered with the point cloud of uniform density. Because of these requirements combined with the types of models produced, these methods are not useful for industrial reconstruction.

### Estimation methods

Estimation methods are required to find the best value of parameters for the model being fitted. The 'best' value is defined in terms of the chosen distance measure. Least squares fitting, M-estimators, least median of squares, simulated annealing and genetic algorithms like Tabu search (Srinivas and Patnaik, 1994) are some of the available techniques.

The most commonly used estimation method is that of the linear least squares. As noted by Press et al. (1988) least-squares fitting is a maximum likelihood estimator of the fitted parameters if the measurement errors are independent and normally distributed with fixed standard deviations. However, the probability of outliers in the Gaussian distribution is so small that their occurrence is not handled robustly; most of the time a few outliers can influence the resulting fit away from the required result. To counter the sensitivity of least squares to outliers, the use of robust estimation techniques is necessary. A survey of the robust parameter estimation techniques, as applied to the problems faced in computer vision, is given by Stewart (1999). It compares various parameter estimation methods on the basis of their breakdown point, influence function and efficiency. It is reported there that different robust estimators provide different trade-offs and none of them solves all problems. An alternative robust approach for the estimation of parameters based on random sampling and called Random Sample Consensus (RANSAC) has been presented by Fischler and Bolles (1981, 1987).

A study of the different estimation methods using different distance measures for conic fitting is presented in Zhang (1997). They report that the orthogonal distance based fitting works much better compared to the algebraic distance fitting, especially in the presence of outliers and noise. They obtained the best results by using the least median of squares estimator with the orthogonal distance. However, as noted by (Stewart, 1999), least median of squares has low statistical efficiency, and the obtained data must be dense for adequate performance.

### Conversion to a CAD representation

The results of surface fitting usually do not present a complete picture because they are based on incomplete point clouds and thus cannot be directly used as a CAD model. Moreover, most industries use CAD standards based either on Boundary representation (B-rep) or Constructive Solid Geometry (CSG) (Mortenson, 1985; Requicha and Voelcker, 1982). A manual or semi-automatic stage of editing is necessary to convert the fitted surface patches to a full CAD model. This stage, called by Várady et al. (1997) ‘global surface fairing’, usually involves extending, clipping, and intersecting the fitted surfaces to get a full B-rep or CSG model.

Geometric constraints encodes a priori information about the geometry of the model and their incorporation into fitting can lead to a significant reduction in the degrees of freedom resulting in better estimation. Some techniques for recovering quadric surface based models by using constraints have been presented by Werghi et al. (1999a). They first extract planar and quadric patches from the segmented range data, analyze them to infer the geometrical constraints like perpendicularity, equality of radius etc. These constraints are then used to improve the fitting results.

As most modeling procedures consist of two separate processes of surface fitting and conversion to CAD model, it is difficult to simultaneously enforce the geometric constraints. Compared to B-rep, CSG provides a powerful, though less general, method to specify both the surface geometry and associated constraints in one package. Based on this observation, techniques for fitting CSG models with constraints to images have been developed by (Ermes et al., 1999; Tangelder et al., 1999, 2003). Currently, there are no reported methods for fitting of the CSG models to point clouds.

Sequeira et al. (1999) and Stamos and Allen (2002) present methods for reconstruction of buildings and cultural heritage sites using range data and images. They use images mainly for texture mapping and visualization.

## 1.1.3 Registration

For successful reconstruction it is necessary to combine range data taken from different viewpoints. Registration is the process where scans are transformed from their local coordinate system to one global coordinate system. The following are some of the reasons which necessitate registration:

- For most objects of interest it is not possible to capture full details of their 3D shape from one scan. Limited FOV of the laser scanner, occlusions from the neighboring objects, and self occlusion are some of the possible reasons. As a result scans from different viewpoints must be combined to get a true and complete picture.

- Even when all the parameters of a surface can be estimated from one scan, combining more scans gives more dense data with better coverage, and thus leads to a better estimation of the object parameters during model fitting.

A problem related to registration is that of view-planning. For faithful reconstruction of any object from the range data it is necessary to take sufficient scans. But how many are sufficient and from where should they be acquired? Certainly, it is desirable to have as detailed a picture as possible, but each scan has an associated price in terms of time and labor involved. Thus it is important to optimize the data acquisition so that a sufficient detail is captured with minimum cost using minimum scans. This problem is addressed by the task of view planning or network design.

The simplest way, which is mostly used for reverse engineering, is to put the object on a turntable and then capture its range data by rotating the setup. Such a solution is practically impossible for any outdoor environment including that of industrial installations. This makes it necessary to do view-planning in advance, using a priori information about the site to be modeled. An occlusion based view-planning approach is described by Maver and Bajcsy (1993). Another possibility is to use multiple sensors which view an object from different sides simultaneously (Eggert et al., 1998a). Such a setup, though effective for industrial inspection and reverse engineering of mechanical parts, is not a good choice for industrial reconstruction.

Effective view planning must ensure that there is a sufficient overlap between different scans so that they can be used for registration. In the absence of sufficient overlap, the resulting registration will be erroneous or in some cases may not be possible at all. Some reference objects can be installed at the site which can be later used as 3D fiducials or control points. However, such methods are cumbersome and may not be feasible for each situation. A comprehensive set of rules for planning the acquisition of laser scans and images for modeling of industrial sites is given by Heuvel (2003). A survey and comparison of different view planning techniques for automated 3D object reconstruction and inspection by means of active, triangulation-based range sensors is given by Scott et al. (2003).

For registration of two scans with at least three known point correspondences closed-form solutions have been given by Horn (1987). However, it is quite difficult to give reliable point correspondences in the point clouds because of their limited sampling density, measurement noise and occlusions. To solve this problem Besl and McKay (1992) proposed the Iterative Closest Point (ICP) method. Starting from a good initial alignment ICP iteratively finds the correspondences by getting the closest points in one scan to the points in the other. Many variations on ICP have been proposed in the literature which differ mainly in the type of distance metric used for finding the closest point. A comparison of these methods was done by Rusinkiewicz and Levoy (2001), where the distance to closest tangent plane was found to give the best performance.

Johnson and Hebert (1997) propose a different method for range registration based on a pose-invariant local neighborhood representation for each point called *Spin image*. The spin image encodes the spatial relationship of a point with its neighbors. An initial list of correspondences is generated by matching spin images of the scans being registered. This list is later refined by ICP.

Lucchese et al. (2002) has presented a method for range registration that operates in the frequency domain through the Fourier transform. As the Fourier transform decouples the rotation and the translation parameters, the presented algorithm exploits this property by doing registration through sequential estimation. The presented results are inferior to ICP, but the method can be used to provide automatic approximate values.

Pottmann et al. (2004) has given a method for registration based on instantaneous kinematics and on the geometry of the squared distance function of the point cloud. The algorithm is shown to have better convergence properties than the ICP.

If both range data and colored images of a site are available, various data fusion methods can be used to improve the registration results. A multi-feature ICP matching algorithm that includes the surface color and the surface orientation information has been proposed in (Schutz et al., 1998). Other approaches for registration making simultaneous use of images and point clouds are reported by El-Hakim et al. (2004); Sequeira et al. (1999); Stamos and Allen (2002).

### 1.1.4 Object recognition

The problem that object-recognition tries to solve can be defined as follows:

- $\mathbf{R}$ is a representation of the object in some n-dimensional space $\mathbb{R}^n$ . It can consist of raw range measurements and/or some features extracted from it.
- $\mathbf{T}$ is a set of transformations which can be applied to $\mathbf{R}$ to generate a transformed and noise-free representation $\mathbf{T}(\mathbf{R})$ in the feature space. This transformation can be affine, projective or some other non-linear transformation.
- $\mathbf{D}$ consists of the representation of the observed data is the same n-D space as $\mathbf{T}(\mathbf{R})$. It will be most probably noisy and will contain occlusions.
- $E$ is a function that gives the similarity between $\mathbf{D}$ and $\mathbf{T}(\mathbf{R})$.

Object recognition tries to estimate the optimal value of $\mathbf{T}$, and the best set $\mathbf{R}$ such that $E(\mathbf{D}, \mathbf{T}(\mathbf{R}))$ is maximized.

This problem is far from trivial because of the following reasons:

- It is not known in advance which of $\mathbf{R}_i$'s are present in the image. There might be multiple instances of one type and/or multiple objects of different types.

- The transformation $\mathbf{T}_i$ for each object is not known. Moreover, it is not necessary that $\mathbf{T}(\mathbf{R_i})$'s are unique, as two different objects in combination with different $\mathbf{T}$'s may produce similar data in the observation space.
- Some part of the object may be occluded by other objects, leading to incomplete observations.
- The presence of noise, missing data and outliers due to the imperfection of the data collection and/or data processing further complicates the process.
- There might be new types of objects present in the scene, which are previously unobserved and thus are not recognizable.

The various approaches in the literature for solving the problem of object recognition and pose-estimation can be categorized into two main categories: Object-based recognition and View-based recognition.

### Object-based recognition

In object based recognition different features are identified in some CAD like representation of the object model and are used for matching. Different features like silhouettes, edges, surface curvatures, local shape, point features etc. can be used for this purpose. The ultimate goal is to come up with a collection of features that are invariant to pose and scale differences and can be robustly estimated from the noisy data, with outliers and incomplete information due to occlusions.

Most systems employing object-based recognition operate in three different stages: training, matching and verification. In the *training stage* all objects are represented as a collection of the selected features in the form of a database. The implementation of this database depends on the types of features selected and the degree of invariance they possess. The second step of the *matching stage* consists of extracting the same features from the captured data and using them to index the database to get a set of strong hypotheses. The best hypothesis is selected as the one that best satisfies a user-defined similarity function. A *verification stage* may follow, which tries to reject false positives. The final performance depends on the selection and extraction of features and how they are used to index the database to get strong hypotheses.

Local features of the surface as given by classical differential geometry (Carmo, 1976) are a natural choice for object recognition, due to their invariance to pose, and robustness in the presence of occlusion. Many approaches have used principal curvatures of 3D data and its derivatives like Gaussian and mean curvature for object recognition.

Another strategy sometimes called *object recognition by model alignment* (Huttenlocher and Ullman, 1990) tries to detect the points of high-curvature on the surface, uses it to select a set of possible hypotheses from the object database, and calculates the pose for each of them. Each of these pose-corrected models is then matched to full range data to select the best representative object.

The type of object-based recognition methods described above have been labeled as *generate-and-test* methods by Olson (2001b), because they select a minimal representation in terms of invariant features which is used to generate hypotheses. Each hypothesis is then tested and validated according to some figure-of-merit. The biggest problem with these approaches is their sensitivity to noise and outliers, because the generation of the hypotheses depends on the successful detection of distinguished features, and a failure to do so means the testing stage is going to fail as well. Even small errors in the localization of the distinguishing features can lead to large errors in the estimated models. To counter these problems, techniques employing error-bounds have been developed. The basic idea is not to require the distinguished features to be fully aligned, but to accept each alignment with its error bound (Alter and Jacobs, 1998; Grimson et al., 1994). By propagating this error bound to the testing stage the probability of miss can be decreased, but at the same time false-positives also increase.

An alternative to generate-and-test paradigm is to use Hough-based method for model extraction.. These parameter space analysis methods originated from the patent of Hough (1962), which was initially meant for the detection of the curves produced by bubble-paths from the images of bubble-chambers.

The basic idea behind the Hough transform is to map each data or feature point to a manifold in the parameter space. This parameter space manifold is usually represented in a quantized manner where each cell acts as an accumulator, collecting votes. The procedure consists of mapping data points to this parameter space and then finding the cells with maximum number of votes. Initial work on the Hough transform was done by Duda and Hart (1972); Rosenfeld (1969). Ballard (1981) extended the Hough transform for the detection of arbitrary shapes giving what he called the *Generalized Hough transform*. He also used the orientation information to speed up the process of localizing the best match in the parameter space.

The Generalized Hough transform technique when applied to the task of object recognition is usually called *pose-clustering*. It consists of initializing an array of bins, indexed by the parameters of object pose, as empty. For each possible match between one image feature and one model feature, poses consistent with that match are determined and votes are cast in the bins corresponding to those poses. Finally, the array is scanned to identify and verify those poses that have received the most votes. The identification of best hypothesis is done by looking for the biggest clusters in the parameter-space, hence the name *pose-clustering*. The cluster analysis usually adopted for this purpose is as simple as multidimensional histogramming.

One weak-point of the Hough transform based methods for object recognition is their prohibitively large computational and memory requirements. These become even more important when the database of models to be matched grows big. Randomization has been used by many researchers as a solution to the problem of computational and space complexity (Bergen and Schweitzer, 1991; Kalviainen et al., 1994).

A hybrid approach for object-recognition that combines both generate-and-test and Hough-transform-based methods is reported by Olson (2001a). The first stage generates possible hypotheses using distinguished-features but the final testing stage uses randomized Hough transform. This method is called RUDR (Recognition Using Decomposition and Randomization).

Another technique similar to the Hough Transform is template matching, its main limitations being the computational complexity and the sensitivity to noise and occlusions. The brute-force approach of template matching consists of translating each possible transformation of the object on the acquired data, calculating a similarity metric, and finding the points where this metric exceeds a certain threshold. The computational requirements of such an implementation are unacceptably high. As a result, most of the literature concerned with this method focuses on efficiency improvements. Various pyramid like multi-resolution methods have been suggested. A hierarchical representation of templates in the form of a tree has been proposed by Ramapriyan (1976). Each parent node of the tree consists of a union of all child nodes, making it a representative node. The success of such an approach depends on the effective tree-like representation of the templates. The parent nodes on each level should be sufficiently different from each other. Similar tree based template matching method has been used by Greenspan (1998); Greenspan and Boulanger (1999) for object recognition. Moreover, hierarchical tree and pyramidal techniques can be combined to get further computational efficiency.

### View-based recognition

The view-based techniques take a completely different approach to the problem of object recognition. They aim to bypass the pose-estimation stage altogether. This is achieved by presenting a pattern recognition system with a set of views captured from all viewpoints during the training stage. Once such a system has been trained, the recognition just consists of presenting the observed data to the system which gives the object corresponding to the best possible match as the result.

Due to the computational complexity of direct matching of incoming samples to each of the training objects, most approaches suggest different ways of compressing the training set so that it can be represented with a minimum number of coefficients. One of the most popular approaches for this dimension reduction is eigen-space representation based on principal component analysis (PCA) (Webb, 1999). Due to the high correlation of the range data of an object captured from different viewpoints, PCA can produce effective dimension reduction. Once trained, the recognition step consists of projecting the observed data onto the eigen-space of each model in the database, and selecting the model that gives the maximum similarity.

Another approach used for object recognition is based on *aspect graphs*, which are data structures which efficiently store information about the views of an object or collection of objects in a given scene. Given a view of an unknown object,

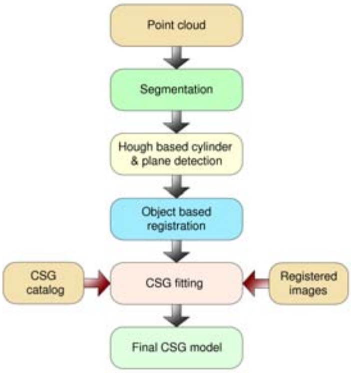


**Figure 1.3:** *Flowchart of the presented modeling pipeline. Initially the point clouds is used as main data source due to better automation possibilities. Images are included at the end to improve estimation of parameters. Fitting of CSG models along with their constraints removes the manual extrapolation and editing stage.*

its identity is established by finding the closest match in a given set of aspect graphs (Bowyer et al., 1993; Cyr and Kimia, 2004; Eggert et al., 1993; Fischer and Steinhage, 1997; Gigus and Malik, 1988). Aspects graphs have also been used for object detection in range images (Morita, 1999).

Campbell and Flynn (1999) coin the term *Eigen surfaces* for the use of PCA for 3D object recognition, thus extending the scope of Eigen faces used for face recognition. They implemented a system that could recognize objects from different viewpoints having different rotation parameters. They reported the results on an object database of manufactured parts.

View-based object recognition approaches require the object to be completely visible and without many outliers. This means that for a scene consisting of multiple objects, a reliable segmentation of the data has to be done as a pre-processing step. Many approaches have been presented to address the problems of missing pixels and occlusions, for example see Amano et al. (1996); Skocaj and Leonardis (2001). Most of these methods replace the step of data-projection onto eigen space basis-vectors by a solution of an over-constrained system of linear equation formed only from the visible pixels. This process is much slower than simple projection but is more robust to the effects of missing information.

Various data fusion approaches combining range data, color images, and a priori information have been suggested in the literature. Strat and Fischler (1991) use monochrome, color, stereo, or 3D range images along with contextual knowledge, making the process of recognition robust to various inconsistencies in the acquired data.

For a survey of different representation schemes, systems and approaches used for 3D object recognition see Jain and Dorai (2000), Besl and Jain (1985), Arman and Aggarwal (1993) and Bennamoun and Mamic (2002).

## 1.2 Objectives and modeling pipeline

From the survey of the previous work presented in the last section we see that although there has been a lot of work done on general 3D reconstruction, there has been little effort made to use the domain specific information from the industrial environments to automate the detection and estimation of the present models. Similarly, the scan registration problem which currently uses point correspondences can be automated for industrial scenes by using corresponding objects for registration. As most industrial environments consist of a small number of well defined objects (planes, cylinders, spheres, tori, and cones) and their combinations, the detection of these objects in each scan followed by an efficient search for correspondences provides an opportunity to automate the process of registration. The combination of simple objects can be flexibly represented as CSG objects whose direct fitting avoids the manual editing stage which is commonly employed for conversion of fitted surfaces to complete B-rep models. Additionally, most of the current modeling approaches use images only for texture mapping or visualization. We show in Chapter 7, that the inclusion of images in fitting can result in superior estimation quality and present methods for this purpose. Similarly, the use of geometric constraints which is quite popular in CAD design has not been fully realized in the domain of 3D reconstruction from point clouds. We present a taxonomy of commonly encountered geometric relationships in CSG models and methods for including nonlinear equality and inequality constraints in the model estimation. Although, the presented methods are limited in the sense that they address a very narrow domain of industrial reconstruction, at the same time this narrow domain allows us to make assumptions about the expected structure which is then exploited for the automation of different steps of 3D reconstruction.

The objectives of this thesis are as follows:

- Development of a segmentation method that does not use curvature or higher order derivatives, and has a small number of configuration parameters which have an intuitive meaning. The tuning of those parameters should result in different trade-offs between under and over-segmentation. Most of the current methods use higher order derivatives whose estimation from noisy point clouds is error prone. Additionally, incorporation of a large number of configuration parameters makes their fine tuning very difficult.

- Development of efficient Hough transform based methods for automatic detection of planes and cylinders in the point clouds. Although the Hough transform is widely used for line detection in images, it is not computationally feasible to directly apply it to higher dimensional detection problems. This necessitates development of efficient and sequential processing methods.

- Development of algorithms for fitting simple models to the point clouds and for the calculation of their approximate values. A comparison of the different approximation schemes for calculating point to body distance for the fitting of CSG objects. As most industrial objects can be efficiently represented as

CSG the fitting methods based on CSG have more potential to decrease the amount of manual editing.

- Development of techniques and methods which allow the use of the detected objects as targets for inter scan registration. As industrial scenes contain lot of well-defined structure, it should be exploited to automate the task of point cloud registration.
- Development of techniques for specifying and including geometric constraints in the model estimation procedure for CSG models. As constraints are used during designing of industrial parts, their inclusion in the fitting procedure results in better estimation quality and makes sure that design intent is enforced during modeling.
- Inclusion of images in the fitting process and an analysis of the resulting improvements in parameter estimation. Images provide a complementary source of information, as they capture the object edges where point clouds are mostly very noisy. The inclusion of images in model fitting should result in better estimation quality.

To achieve the above outlined objectives we developed a modeling strategy which can be best represented as a pipeline, shown in the form of a flowchart in Figure 1.3. We start from the point cloud as the main data source because due to the explicit 3D information it provides a richer source for automatic detection and modeling. The first step in our processing pipeline is segmentation of the point cloud, which divides the point cloud into a set of disjoint groups or segments. Our segmentation method is based on "smoothness constraint", which detects smoothly connected areas without over-segmentation. Each segment is processed through the Hough transform for plane and cylinder detection. The segmentation is necessary because the size and resolution of the Hough transform depends on the extent of the data. By sub-dividing a given point cloud into a set of mutually disjoint regions we ensure that a good resolution is obtained at moderate computational cost. To refine the parameters of the objects detected through the Hough transform, a fitting stage based on non linear least squares follows. Each scan individually passes through these stages of segmentation and object recognition.

The scans are registered automatically by searching for the correspondences of the objects detected in the previous stage, which are used as targets for model based registration. This automatic search is optimized by the propagation of geometric constraints. The approximate registration is followed by a stage of integrated adjustment, which extends the concept of bundle adjustment from traditional photogrammetry (Atkinson, 1996) to the domain of point cloud processing. In this stage registration parameters of the scans, and the shape and the pose parameters of the models are simultaneously estimated.

Once the scans have been registered, we fit models from a catalog consisting of CSG model templates. This fitting also utilizes registered imagery if it is available. The inclusion of images gives better estimation results, especially as the information from well-defined edges in the images helps to better resolve the geometry of bounded objects. In this regard images provide a complementary source of information (see Figure 1.2 (d-e)).

## 1.3 Outline of the thesis

In Chapter 2 we present a method for the segmentation of the point clouds, which avoids over-segmentation while partitioning the input data into mutually disjoint, smoothly connected regions. It uses a criterion based on a combination of surface normal similarity and spatial connectivity, which we call *smoothness constraint*. As we do not use surface curvature our algorithm is less sensitive to noise. Moreover, there are only a few parameters which can be adjusted to get a desired trade off between under- and over-segmentation.

Chapter 3 provides details of the extensions of the Hough transform for automatic plane and cylinder detection in the point clouds. For plane detection the Hough transform is three dimensional. For the cylinder detection the direct application of the Hough transform requires a 5D Hough space, which is quite impractical because of its space and computational complexity. To resolve this problem we present a two-step approach requiring a 2D and 3D Hough space. In the first step we detect strong hypotheses for the cylinder orientation. The second step estimates the remaining three parameters of the cylinder i.e. radius and position. The results of applying the presented techniques to some industrial point clouds are also presented.

The problem of fitting models like planes, cylinders, spheres, cones, tori and CSG models to point clouds is addressed in Chapter 4. The methods for the calculation of approximate values for these models are also given. For the fitting of CSG models we present three different methods for approximating the orthogonal distance, which are compared based on speed and accuracy. The fitting approach presented in this chapter uses only linear constraints whose solution is incorporated into an unconstrained least squares approach. This provides an easy, though less general, approach of solving for constraints.

Chapter 5 presents methods for using modeled objects in scans as targets for registration. As the available geometric structure is used, there is no need to place artificial targets. We present two different methods for this purpose called Indirect and Direct methods. The Indirect methods is a quick way to get approximate values while the Direct method is then used to refine the approximate solution. The final stage is called Integrated Adjustment which leads to simultaneous least squares estimation of model shape and scan registration parameters. This is an extension of the standard bundle adjustment to the domain of point cloud processing. The presented methods are compared to ICP and show superior performance in terms of the quality of parameter estimation.

In Chapter 6 we present techniques for automatically finding the corresponding objects for registration of scans. The presented techniques are based on constraint propagation which use the geometric information available from the previously made correspondence decision to filter out the possibilities for future correspondences. We compare the improvements obtained by incorporating the presented constraints with the direct application of exhaustive and RANSAC based search.

As discussed above, images provide a complementary source of information as they contain well-defined edges of the bounded objects. Although the point clouds capture the surface information very accurately on smooth areas, the measurement quality deteriorates in the areas where the surface normal changes rapidly. This is especially true for the object edges. We present methods for the fitting of CSG models to a combination of point clouds and images in Chapter 7. We also present techniques for the specification of geometric constraints between sub-parts of a CSG tree and their inclusion in the model estimation process. A taxonomy of commonly encountered geometric constraints and their mathematical formulation is also given. In contrast to Chapter 4 here we use constrained optimization methods which make the fitting approach more general and flexible. Fitting experiments are presented to highlight the improvements obtained in the estimation quality by using images and point clouds simultaneously.

Finally, we present a summary of the achievements with some directions for future research in Chapter 8.

Chapter 2

# Segmentation using smoothness constraint

> But what is classification but the perceiving that these objects are not chaotic, and are not foreign, but have a law which is also the law of the human mind?
>
> Ralph Waldo Emerson (1803-1882)

> … you must acquire and beget a temperance that may give it smoothness.
>
> William Shakespeare (1564-1616)

## 2.1 Introduction

### 2.1.1 Problem statement

Segmentation is the process of labeling each measurement in a point cloud, so that the points belonging to the same surface or region are given the same label. For the problem of industrial reconstruction the point cloud is usually acquired using a laser scanner. Unlike structured light based instruments which provide $2\frac{1}{2}$D data, most of the laser scanners provide an unstructured point cloud. Even in the case of $2\frac{1}{2}$D range images, once two or more such images have been registered, the resulting data loses its $2\frac{1}{2}$D character and has to be represented as an unstructured 3D point cloud. Based on these observations the presented method would work only with unstructured point clouds, and other data representations can be easily converted to this format if required.

We follow the formal definition of segmentation as given in Hoover et al. (1996), which is as follows:

Let $R$ represent the entire point cloud. Segmentation can be viewed as a process that partitions $R$ into sub-regions $R_i$ such that:

1. $\cup_{i=1}^{n} R_i = R$
2. $R_i$ is a connected region, $i = 1, 2, \cdots n$
3. $R_i \cap R_j = \Phi$ for all $i$ and $j$ and $i \neq j$ , where $\Phi$ is the null set
4. $P(R_i) =$ TRUE for $i = 1, 2, \cdots, n$ and
5. $P(R_i \cup R_j) =$ FALSE for $i \neq j$ and $R_i$ and $R_j$ are adjacent where $P(R_i)$ is a logical predicate over the points in set $R_i$.

The predicate $P(R_i)$ defines the measure of similarity that groups the points in one region and separates them from the points in other regions. In the presented method the predicate is based on smoothness constraint, requiring that the normals of the points in one region do not change too rapidly and that the points are spatially connected.

### 2.1.2 Previous work

A literature survey of different segmentation methods has been presented in Chapter 1. Here we will discuss the limitations of these methods and outline the motivation for developing a new approach.

A comparison of methods for finding planar segments in range images was done by Hoover et al. (1996). Although planar surfaces are quite frequently found on industrial sites, any segmentation strategy that uses planes as the only available model will result in extreme over-segmentation in curved areas like pipes, spheres and bends.

Based on the comparison framework of Hoover et al. (1996), two methods for segmentation of range images into curved regions were compared by Powell et al. (1998). The first segmentation method of Besl and Jain (1988b) (BJ method) has two stages. The first stage of coarse segmentation is based on estimating the mean and Gaussian curvature for each point and using their signs for classification into 8 different surface types. This rough segmentation is refined by the second step of region growing, which is based on fitting bivariate polynomial surfaces. The comparison found the BJ method to result in severe over-segmentation even on very simple scenes with low noise. The major reason for this failure was the error in the estimation of principal curvatures from the noisy range data. The BJ segmentation method has 38 different parameters, 10 of which were iteratively optimized to get the best possible results. The speed of this method was very slow, and even on range images which do not have a high cost for searching the neighborhood points, it took more than 6 hours.

The second segmentation method used for the comparison was by Jiang et al. (1996) (UB method). This method also consists of two stages. In the first stage the scan lines of the range image are segmented into a set of curves by using a splitting method. In the second stage these edges are grouped together to make surfaces. This method has 10 parameters and the comparison found it to perform much better than the BJ method, both in terms of time (30 seconds compared to 6 hours of the BJ method) and the quality of results.

Looking at the comparison, the UB method would be a good choice for the segmentation of industrial scenes, but there are some serious limitations. First of all, the method is based on the grouping of scan lines which do not exist for unstructured point clouds. For airborne laser scanner data similar scan lines created by collecting and joining points in a tubular volume have been used by Sithole and Vosselman (2003) for segmentation, but there the data is $2\frac{1}{2}$D . Extensions of this idea for 3D point clouds would require choosing a few preferred directions for scan-lines, making the results of segmentation orientation dependent.

### 2.1.3 Problems with existing methods

We observed the following problems with existing segmentation techniques as applied to our problem of processing industrial point clouds:

1. Many approaches are tailored only for planar surfaces, which are too limiting for industrial scenarios.
2. Although principal curvature based approaches can handle curved objects, the unreliable estimation of the curvature from noisy point clouds leads to high rates of over-segmentation. Furthermore, the objects like torus and sphere are always over-segmented because they are not one of the 8 different surface classes identifiable based on the signs of the principal curvatures. The sensitivity of curvature estimates from range data has been analyzed in (Trucco and Fisher, 1995), where it is suggested that at least for planar segmentation of range data principal curvatures should not be used.
3. Many segmentation methods have a large number of parameters, whose meaning and effects on final segmentation are not always clear. Most of the comparisons used separate iterative optimization methods to find the best set of parameters.
4. Most of the methods are tailored for application to $2\frac{1}{2}$D range images. Sometimes their extension to 3D unstructured point clouds is quite simple like replacement of 8-neighbors with k-nearest neighbors. But in other cases, like defining scan lines for 3D point clouds, there is no straight forward extension, and most approaches introduce new limitations.
5. There are some model-based approaches to segmentation which segment and recognize surface types at the same time (Marshall et al., 2001). These approaches do not fit in our pipeline of separate segmentation and object

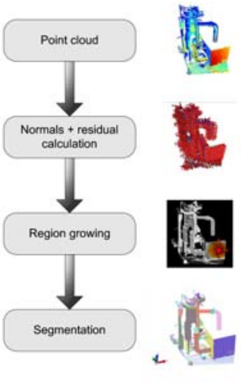


**Figure 2.1:** *Flowchart of the segmentation algorithm. For a given point cloud surface normals are estimated through plane fitting to the neighborhood of each point. The residual of plane fitting is used as approximation to local curvature. The region growing uses both normal and their residuals for enforcing* smoothness constraint

recognition through Hough transform and thus cannot be applied to our problem.

### 2.1.4 Objectives and motivation

Noting these weaknesses we decided to develop a simple segmentation strategy that follows the following guidelines:

1. We will assume a raw unstructured 3D point cloud as the input to the algorithm. Although the assumptions about structure of data (range image, TIN etc) can make the job of neighbor search faster, they at the same time make the algorithm less general purpose.
2. We will use only surface normals as they can be reliably estimated even in the presence of noise (provided the neighborhood is sufficiently big to average out effects of noise).
3. While deciding between over and under-segmentation, we will prefer under-segmentation. In our modeling pipeline the segmentation is followed by the stage of object recognition, which processes each segment separately. That stage can detect multiple objects in one segment (under segmentation), but if an object is split into multiple segments (over-segmentation), its detection and correction would be more difficult.
4. The algorithm should have a low time and space complexity. Furthermore, there should be a few parameters having a physically intuitive meaning.

## 2.2 Segmentation algorithm

As stated earlier the basic purpose of the presented segmentation algorithm is to subdivide the input point cloud into meaningful subsets, while avoiding both

**Algorithm 1** Segment a given point cloud using smoothness constraint

```
**Inputs**: Point cloud = $\{\mathbf{P}\}$, point normals $\{\mathbf{N}\}$, residuals $\{\mathbf{r}\}$, neighbor finding function $\boldsymbol{\Omega}(.)$, residual threshold $r_{th}$, angle threshold $\theta_{th}$
**Initialize** Region List $\{\mathbf{R}\} \leftarrow \Phi$, Available points list $\{\mathbf{A}\} \leftarrow \{1 \cdots P_{count}\}$
**while** $\{\mathbf{A}\}$ is not empty **do**
    Current region $\{\mathbf{R}_c\} \leftarrow \emptyset$, Current seeds$\{\mathbf{S}_c\} \leftarrow \Phi$
    Point with minimum residual in $\{\mathbf{A}\} \rightarrow P_{min}$
    $P_{min} \overset{\text{insert}}{\rightarrow} \{\mathbf{S}_c\}\&\{\mathbf{R}_c\}$
    $P_{min} \overset{\text{remove}}{\rightarrow} \{\mathbf{A}\}$
    **for** $i = 0$ to **size**$(\{\mathbf{S}_c\})$ **do**
        Find nearest neighbors of current seed point $\{\mathbf{B}_c\} \leftarrow \boldsymbol{\Omega}(\mathbf{S_c\{i\}})$
        **for** $j = 0$ to **size**$(\{\mathbf{B}_c\})$ **do**
            Current neighbor point $P_j \leftarrow \mathbf{B}_c\{j\}$
            **if** $\{\mathbf{A}\}$ contains $P_j$ and $\cos^{-1}(|\langle \mathbf{N}\{\mathbf{S}_c\{i\}\}, \mathbf{N}\{P_j\}\rangle|) < \theta_{th}$ **then**
                $P_j \overset{insert}{\rightarrow} \{\mathbf{R}_c\}$
                $P_j \overset{remove}{\rightarrow} \{\mathbf{A}\}$
                **if** $\mathbf{r}\{P_j\} < r_{th}$ **then**
                    $P_j \overset{insert}{\rightarrow} \{\mathbf{S}_c\}$
                **end if**
            **end if**
        **end for**
    **end for**
    Add current region to global segment list $\{\mathbf{R}_c\} \overset{insert}{\rightarrow} \{\mathbf{R}\}$
**end while**
Sort $\{\mathbf{R}_c\}$ according to the size of the region.
**Return** $\{\mathbf{R}_c\}$
```

under and over-segmentation with a preference for under-segmentation in case of a dilemma.

The segmentation method has the following stages:

- Normal and residual estimation
- Region growing

The details of these steps are given in Algorithm 1 and further explained below. See also Figure 2.1.

### 2.2.1 Normal estimation

The normal for each point is estimated by fitting a plane to some neighboring points (Figure 2.2(c)). This neighborhood can be specified in two different methods.

**K nearest neighbors (KNN)** In this method for a given point we select the k points from the point cloud having the minimum distance. The distance metric used can be Euclidean, Manhattan, or any other distance metric obeying the triangle inequality.

As the number of points k is fixed, the method adapts the area of interest (AOI) according to the point density. Assuming that the point density is an indicator of the measurement noise (which is usually the case as for a given laser scanner the density falls down inversely with the distance and the angle of incidence), this results in overall better estimation of the normals as a bigger AOI is used in the areas of lower point density (Figure 2.2(a) and 2.2(b)). Moreover, this method always uses the given number of points and avoids degenerate cases (e.g. a point having no neighbors).

Search for KNN can be optimized using different space partitioning strategies like k-d trees (Arya et al., 1998; Goodman and O'Rourke, 1997).

**Fixed distance neighbors (FDN)** This method uses a given fixed AOI, and for each query point, selects all the points within this area. The distance metric used is usually Euclidean but can be changed similar to KNN. For FDN search the number of points changes according to the density of the point cloud. As the number of points is directly proportional to the density of the points in the neighborhood, this method does not have the adaptive behavior of KNN.

Compared to KNN, here the number of points is less in the areas of low density (high noise) and as a result the estimation of the normals is on the whole contains more noise.

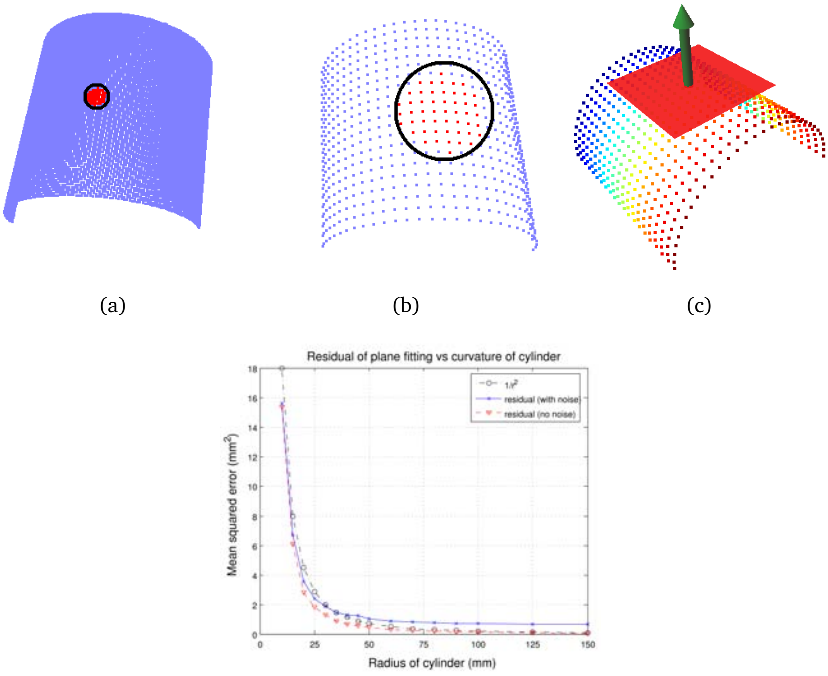
This method is more suitable if the density of the points does not change a lot through out the data. Similar to KNN there are optimized methods for doing FDN searching (Goodman and O'Rourke, 1997; Willard, 1985)

One of the above described methods for neighborhood search are also used during the stage of region growing (Section 2.2.2).

### Plane fitting

To fit any surface to a set of given points, in a least squares sense, we want to find that set of parameters that minimizes the sum of squares of the orthogonal distances of the points from the estimated surface. In general this is a nonlinear least squares problem, but as shown below, in case of planes this can be reduced to an eigenvalue problem.

The plane can be parameterized with its normal $\mathbf{n} = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}$, and its distance from the origin $\rho$. This is also called Hesse normal form of the plane. The distance of any given point $\mathbf{p} = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}$ from the plane is given by $\mathbf{n} \cdot \mathbf{p} - \rho$ provided $\mathbf{n} \cdot \mathbf{n} = 1$. This is a constrained problem and can be solved using Lagrange multipliers.



(a) (b) (c)

(d)

**Figure 2.2:** *(a-b)Adaptive change in selection area for k-neighbors for different point densities (a) hight density, 50 KNN (b) low density, 50 KNN (c) Normal estimation by fitting a plane to the points in the neighborhood (d)Residual of the plane fitting gives an approximation to the local surface curvature*

Given a set of $k$ points in 3D $\{\mathbf{P}\}$ belonging to the plane the Lagrangian function is given by:

$$\Phi = (\mathbf{n} \cdot \mathbf{p} - \rho)^2 - \lambda(\mathbf{n} \cdot \mathbf{n} - 1) = 0 \tag{2.1}$$

$$(n_x p_x + n_y p_y + n_z p_z - \rho)^2 - \lambda(n_x^2 + n_y^2 + n_z^2 - 1) = 0 \tag{2.2}$$

As the unknowns are $n_x$, $n_y$, $n_z$ and $\rho$, the solution can be found by solving the system of equations given by $\frac{\partial \Phi}{\partial n_x} = 0$, $\frac{\partial \Phi}{\partial n_y} = 0$, $\frac{\partial \Phi}{\partial n_z} = 0$ and $\frac{\partial \Phi}{\partial \rho} = 0$. This gives us a direct solution for $\rho$ in terms of $\mathbf{n}$. i.e.

$$\rho = -\mathbf{n} \cdot \bar{\mathbf{p}} \tag{2.3}$$

Where

$$\bar{\mathbf{p}} = \frac{1}{k} \sum_{k} \begin{pmatrix} p_x & p_y & p_z \end{pmatrix} \tag{2.4}$$

Substituting Equation 2.3 in the partial derivatives gives us the following system of equations.

$$\sum_k \begin{pmatrix} (p_x-\bar{p_x})^2 & (p_x-\bar{p_x})(p_y-\bar{p_y}) & (p_x-\bar{p_x})(p_z-\bar{p_z}) \\ (p_x-\bar{p_x})(p_y-\bar{p_y}) & (p_y-\bar{p_y})^2 & (p_y-\bar{p_y})(p_z-\bar{p_z}) \\ (p_x-\bar{p_x})(p_z-\bar{p_z}) & (p_y-\bar{p_y})(p_z-\bar{p_z}) & (p_z-\bar{p_z})^2 \end{pmatrix} \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \lambda \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \tag{2.5}$$

$$\sum_{i=0}^{k} (\mathbf{p}_i - \bar{\mathbf{p}})^T (\mathbf{p}_i - \bar{\mathbf{p}}) \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \lambda \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \tag{2.6}$$

$$\mathbf{A}\mathbf{n} = \lambda \mathbf{n} \tag{2.7}$$

The problem in Equation 2.7 is an eigenvalue problem, with the minimum solution given by the eigenvector of $\mathbf{A}$ corresponding to its minimum eigenvalue. The eigenvalue gives the residual of the plane fitting. Given a solution for $\mathbf{n}$ of the plane $\rho$ can be calculated using Equation 2.3.

#### Residual as approximate curvature

The residual in the plane fitting can arise either from noise or from nonconformity of the neighborhood of a point to the planar model. The second case hints that the residual can be used to find areas of high curvature. Of course we do not get the principal curvatures and their direction from this approximation, but still the edges and the areas of high surface normal variation can be detected based on high residual values of plane fitting.

To check the relationship between curvature and residual of plane fitting we generated data consisting of cylinders of different radii. The normals for these cylinders were estimated by fitting planes to 40 k-nearest neighbors, and then plotted against $\frac{1}{r^2}$ (Figure 2.2(d)). There we see that for the case of no noise the residuals are quite similar to $\frac{1}{r^2}$ except a difference of scale. In the presence of noise the trend remains the same but in addition to a scale factor there is also a shift related to the amount of noise. This supports our idea of using residuals of plane fitting as indicator of areas of high curvature. The regions of high curvature are detected by introduction of $r_t h$ in Algorithm 1.

In Figure 2.4(a), 2.4(e), 2.4(g) we show the residual of plane fitting as color. As expected the areas on edges and points of high curvature have higher residuals.

### 2.2.2 Region growing

The next step in the segmentation process is region growing. This stage uses the point normals and their residuals, in accordance with user specified parameters to group points belonging to the smooth surfaces. This grouping tries to avoid over-segmentation at the cost of under-segmentation.

This stage is based on the enforcement of these two constraints.

**Local connectivity** The points in a segment should be locally connected. This constraint would be enforced by using only the neighboring points (through KNN or FDN) during region growing.

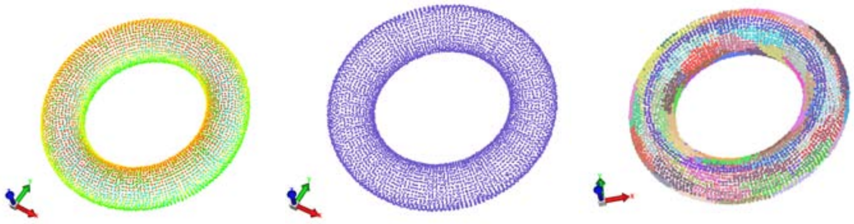
**Surface smoothness** The points in a segment should locally make a smooth surface, whose normals do not vary "too much" from each other. This constraint would be enforced by having a threshold ($\theta_{th}$) on the angles between the current seed point and the points added to the region. Additionally, a threshold on residual values $r_{th}$ makes sure that smooth areas are broken on the edges.

The process of region growing proceeds in the following steps.

1. Specify a residual threshold $r_{th}$. Alternatively, calculate this threshold automatically using a specified percentile of the sorted residuals ($95+\%$ can be a representative number).
2. Define a smoothness threshold in terms of the angle between the normals of the current seed and its neighbors. If the smoothness angle threshold is expressed in radians it can be enforced through dot product as follows $\|\mathbf{n_p} \cdot \mathbf{n_s}\| > cos(\theta_{th})$. As the direction of normal vector has a $180^o$ ambiguity we have to take the absolute value of the dot product.
3. If all the points have been already segmented go to step 7. Otherwise select the point with the minimum residual as the current seed.
4. Select the neighboring points of the current seed. Use KNN or FDN with the specified parameters for this purpose. The points that satisfy condition 2 add them to current region. The points whose residuals are less than $r_{th}$ add them to the list of potential seed points.
5. If the potential seed point list is not empty, set the current seed to the next available seed, and go to step 4.
6. Add the current region to the segmentation and go to step 3.
7. Return the segmentation result

Region growing tries to group points belonging to smooth surface patches together. Although, we want to avoid over-segmentation but still we do not want the whole point cloud coming out as one segment. The inclusion of residual threshold ($r_{th}$) makes sure, that we can strike a balance between the above mentioned extremes. As $r_{th} \rightarrow 0$ we go towards more segments with the extreme case being each point belonging to one segment. Similarly as $r_{th} \rightarrow \infty$ we have less segments and the extreme case of the whole point belonging to one segment.

We can differentiate between the following cases which may lead to the start of a new segment during the process of region growing.



(a) (b) (c)

**Figure 2.3:** *Comparison of segmentation for a toroidal surface (a) point cloud (b) segmentation using presented approach (c)curvature based segmentation*

**Step edge** A step edge is defined by two planes which have the same orientation but different offset from the origin. The segmentation algorithm leads to their separation provided the offset between planes is greater than the AOI for the neighborhood search. For KNN this depends both on the value k and point density, while for FDN it is equal to the fixed distance specified by the user.

**Curvature edge** A curvature edge is defined by the intersection of two surfaces, whose surface normal at the intersection line make an angle greater than the given threshold ($\cos^{-1} \mathbf{n}_1 \cdot \mathbf{n}_2 > \theta_{curv}$). An example of such an edge would be the edge coming from the intersection of the two planar sides of a box.

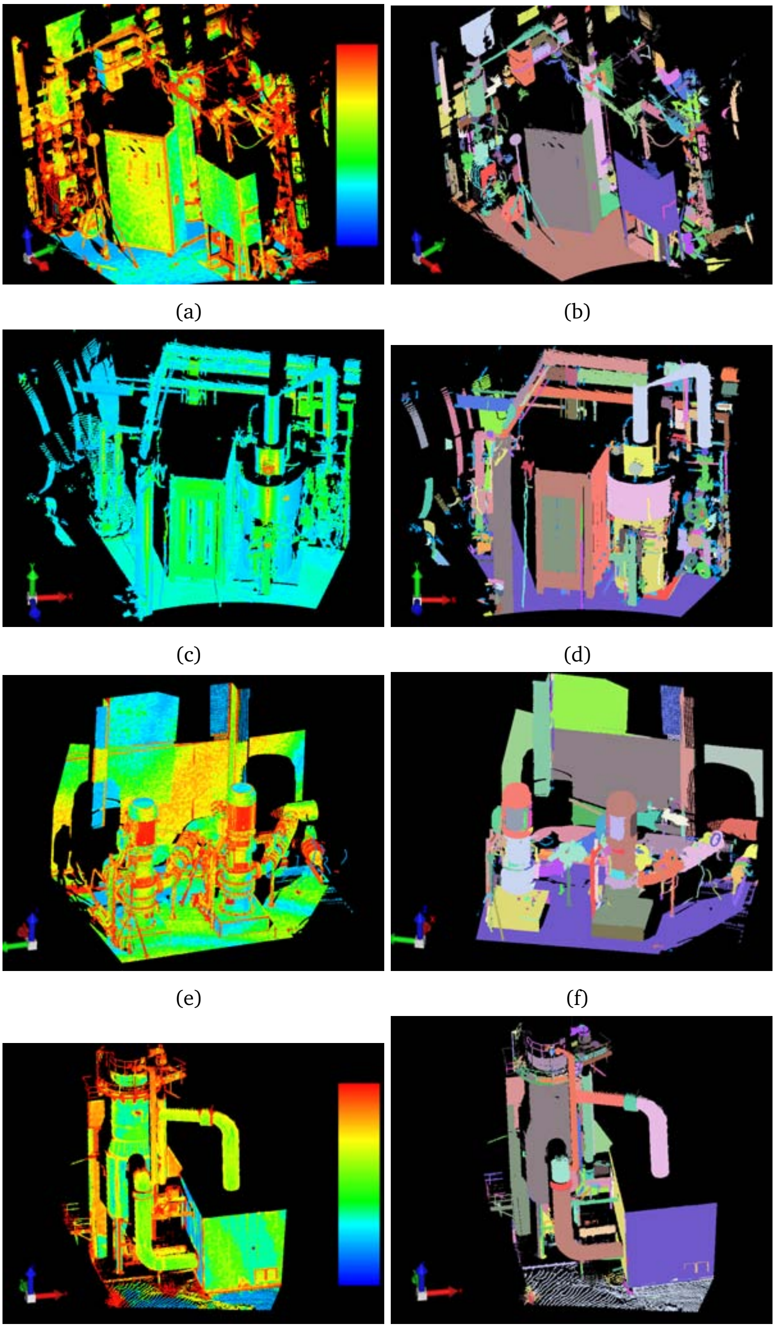
The surfaces on both side of the edge would be segmented because of the smoothness constraint ($\theta_{th}$). Additionally the points on the edge would be marked unsuitable for inclusion in the next generation seeds, as they will have residuals greater than $r_{th}$.

The effectiveness of the method to detect smoothly varying surface patches is shown best in Figure 2.3. While the method of curvature based segmentation leads to high over-segmentation (Figure 2.3(c)), our method divides the data in only one segment (Figure 2.3(b).

## 2.3 Results

The presented algorithm was applied to a set of point clouds acquired from four industrial sites. The results are shown in Figure 2.4. For these results the $\theta_{th}$ was set to $15^o$ and 30 nearest neighbors were used ($k = 30$); $r_{th}$ was automatically calculated by the $98^{th}$ percentile of the plane fitting residuals. In the results we see both goals of grouping smooth areas and avoiding over-segmentation have been successfully achieved. There are areas where large under-segmentation occurred, but it can be explained based on the values the parameters $\theta_{th}$ and $r_{th}$.

For example in Figure 2.4(d) a whole U-section of pipe is segmented as one region,



(a) (b)

(c) (d)

(e) (f)

(g) (h)

**Figure 2.4:** *Results of segmentation (a) residuals of data set 1 (b) segmentation 1 (c) data set 2 (d) segmentation 2 (e) residuals of data set 3 (d) segmentation 3 (e) residuals 4 (f) segmentation 4*

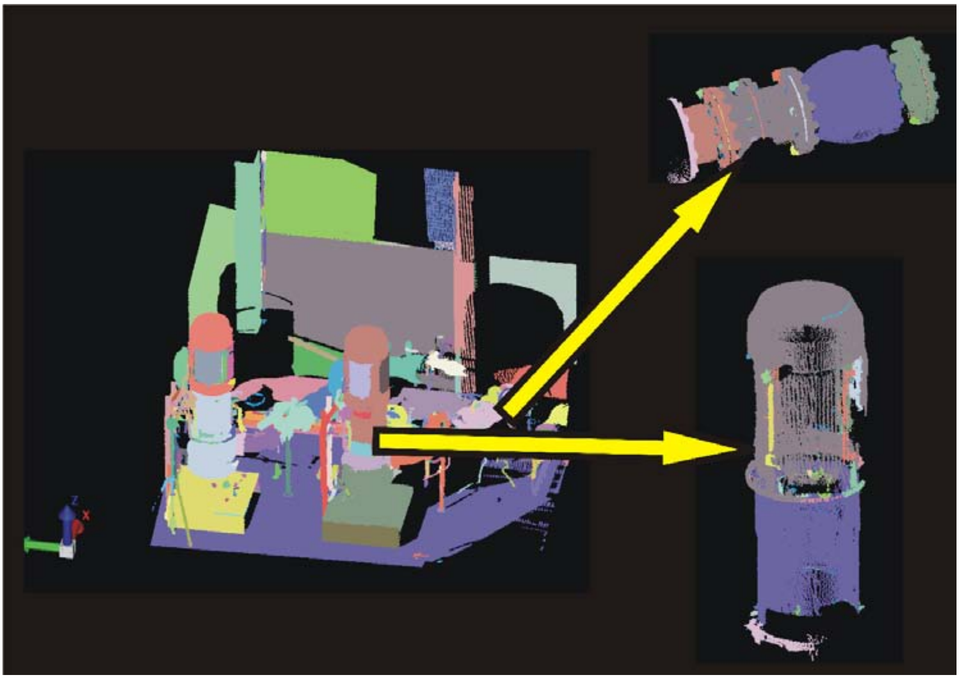


**Figure 2.5:** *Effects of changing $r_{th}$ on the final segmentation. Two segments are re-segmented but with lower $r_{th}$ resulting in more segments*

because it is smoothly connected. Similarly in Figure 2.4(h) the L-junctions of pipes have been grouped as one region rather than being split into two pipes and one curve.

As for the presented results the residual threshold $r_{th}$ was calculated using the percentile method, it leads to data dependent values. In Figure 2.5 we took two segments from the results of Figure 2.4(f) and segmented them again. As now the data is more limited the threshold $r_{th}$ is lower and more strict. This leads to segmentation having more regions along with some over-segmentation. Thus by choosing a proper value for $r_{th}$ the required balance between under and over-segmentation can be achieved.

## 2.4 Conclusions

A segmentation algorithm for dividing a given unstructured 3D point cloud into a set of smooth surface patches has been presented. The algorithm uses only surface normals as a measure of local geometry, which are estimated by fitting a plane to the neighborhood of the point. As fitting of higher order surfaces to noisy point clouds is quite error prone, we approximate the local curvature by the residual of plane fitting. The presented method is a region growing based approach using connectivity and surface smoothness as the guiding principles. The method has two parameters ($\theta_{th}$ and $r_{th}$), which have intuitively clear meaning. Both k nearest neighbor and fixed distance neighbor variations of the algorithm are possible, although we prefer KNN as it adapts its area of interest based on point density. The results on point clouds acquired from industrial sites were presented that show the effectiveness of the method and its tendency to prefer under-segmentation to over-segmentation.

In the next chapter we will use the results of segmentation from the presented method. Each segment will be processed by the Hough transform for the detection of planes and cylinders. Processing each segment separately rather than the full point cloud will provide better speed. Moreover, the quality of the object detection will improve due to a better localization resulting from a reduction in the search area.

Chapter 3

# Object recognition

> In order to begin an analysis, there must already be a synthesis present in the mind.
>
> Johan Huizinga (1872-1945)

> Art is the imposing of a pattern on experience, and our aesthetic enjoyment is recognition of the pattern.
>
> Alfred North Whitehead (1861-1947)

## 3.1 Introduction

Recent advances in 3D scanning technologies have made possible high-speed acquisition of dense and accurate point clouds at moderate costs (Blais, 2004; Laser scanner survey, 2005). The explicit geometric information available from these point clouds can be used to automate the 3D reconstruction process, which has been largely manual till now. This is especially true for the reconstruction of the industrial sites as due to their man-made origin the presence of well-defined CAD primitives can be expected. As-built modeling of the industrial sites is required for documentation, planning, training, and for various emerging technologies that use Virtual and Augmented reality for different services (STAR, 2004). A high degree of automation in 3D reconstruction should benefit all of these application areas.

Planes and cylinders are two of the most important geometric primitives found on industrial sites. 85% of objects found in industrial scenes can be approximated by planes, spheres, cones and cylinders (Nourse et al., 1980; Petitjean, 2002). This percentage rises to 95% if toroidal surfaces are also included in the set of available primitives (Requicha and Voelcker, 1982).

In this chapter we will present extensions of the Hough transform for the automatic detection of planes and cylinders in point clouds. Although for plane detection this extension is quite straight forward, for cylinder detection the direct application of the Hough transform has prohibitively high computational and space complexity. This complexity arises due to the five parameters of a cylinder that necessitate the use of a five dimensional Hough space. We attack this problem by the methodology of divide and conquer and split the detection process in two separate stages. The first stage requires a 2D and the second stage a 3D Hough space. This breakdown solves the space complexity problem while retaining the advantages of the Hough transform like robustness to outliers and the ability to detect multiple instances.

The rest of the chapter is structured as follows. The details of the Hough transform for plane detection are given in Section 2. The Hough transform for cylinder detection is presented in Section 3. Section 4 gives details of the stage of hypothesis verification and ambiguity resolution needed for differentiating between the planes and the cylinders when both are being detected at the same time. Some results on point clouds of industrial sites are presented in Section 5. We give some concluding thoughts in Section 6.

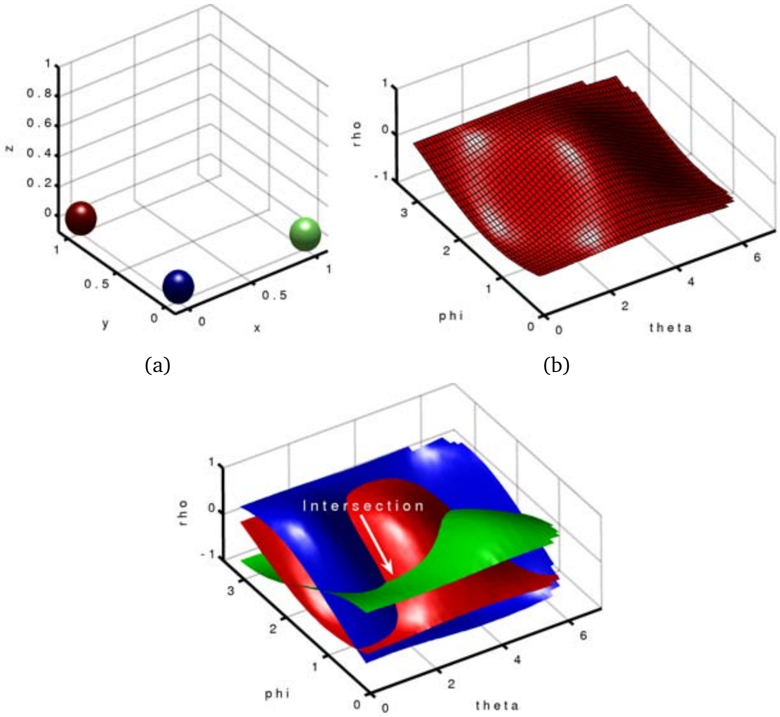
Both Hough transforms require an efficient method for an unbiased and uniform area sampling of the orientation space in $\mathbb{R}^3$ . We present an approximate method for solving this problem in appendix A.

## 3.2 Plane detection

The classical Hough transform for line detection in images is based on the slope-intercept formulation of a line i.e. $y = mx + c$, where $(x, y)$ is a point on the line, $m$ is the slope and $c$ is the y-intercept (Hough, 1962). The Hough transform proceeds by discretizing $m$ and $c$ and for each image point $p = (x_p, y_p)$ incrementing all cells $i$ and $j$ which satisfy $c_i = y_p - m_j x_p$. A big accumulator value in the Hough space gives the hypotheses for lines. There is one major weakness with this parametrization. As the valid range of $m$ is from $-\infty$ to $+\infty$, which cannot be properly discretized, the detection gets poorer as lines become vertical.

This problem for line detection can be solved by parameterizing the line by its perpendicular distance from the origin expressed in the polar coordinates i.e. $x \cos\theta + y \sin\theta = r$, where $\theta$ is the angle of normal direction with the x-axis and $r$ is the perpendicular distance. Using this formulation, the Hough space is still 2D but as $0 \leq \theta < \pi$, we do not have the discretization problem of the classical method described above. Consequently, this formulation enables the unbiased detection of vertical lines.

The extension of classical Hough transform from 2D to 3D for plane detection is quite straight forward. A plane is represented by its explicit equation $z = a_x x + a_y y + d$. Here, we have a 3D Hough space corresponding to $a_x$, $a_y$ and $d$. This extension suffers from the same problems as its 2D counter part i.e., near-horizontal planes can be reliably detected, while the performance deteriorates as planar direction becomes vertical (big values of $a_x$ and $a_y$ amplify the noise in the data). This formulation of the plane has been used for the detection of planes in the



(a)

(b)

(c)

**Figure 3.1:** *Voting in the Hough space for plane detection (a) Input point cloud (b) The Hough space for one point (c) The Hough space for three points, with the intersection of the three voting manifolds giving the parameters of the plane.*

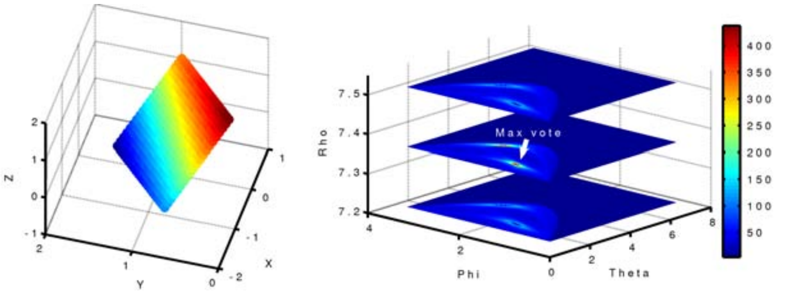
point clouds acquired from airborne laser scanning (Vosselman, 1999; Vosselman and Dijkman, 2001). As there the data is $2\frac{1}{2}$D and most of the planes are non-vertical this scheme works quite well.

For industrial scenes we cannot make the assumption of only non-vertical planes. The solution for unbiased planar detection in 3D is quite similar to the one for 2D. We parametrize the plane by its normal direction $\hat{\mathbf{n}} = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}$ and its perpendicular distance form of the origin $\rho$. This is also called Hesse form of the plane. As there is a constraint on the magnitude of the normal of the plane i.e. $\|\mathbf{n}\| = 1$, there are only three degrees of freedom left.

As an unconstrained representation with the minimum number of parameters is more efficient for the Hough transform we will use spherical coordinates of a unit sphere $(\theta, \phi)$ for representing the unit normal $\hat{\mathbf{n}}$.

$$\hat{\mathbf{n}} = \begin{pmatrix} \cos\theta\sin\phi & \sin\theta\sin\phi & \cos\phi \end{pmatrix} \qquad 0 \leq \theta < 2\pi \quad 0 \leq \phi \leq \pi \tag{3.1}$$

Here we have a 3D Hough space consisting of $\theta$, $\phi$ and $\rho$. Each given point



(a) (b)

**Figure 3.2:** *Discretized Hough space for the points in a plane (a) Input point cloud (b) 2D slices through the 3D Hough space for three different $\rho$ values. The slice in the middle has the highest accumulator value that corresponds to the correct parameters of the detected plane.*

$(x_p, y_p, z_p)$ in the input point cloud votes for all bins $\theta_i$, $\phi_j$ and $\rho_k$ which satisfy

$$\rho_k = x_p \cos\theta_i \sin\phi_j + y_p \sin\theta_i \sin\phi_j + z_p \cos\phi_j \tag{3.2}$$

In Figure 3.1(b) we show the shape resulting in the Hough space from the voting of a single point. In Figure 3.1(c) three such shapes resulting from three input points intersect, and the intersection gives the parameters of the plane. As each plane $(\hat{\mathbf{n}}, \rho)$ can be equivalently represented by $(-\hat{\mathbf{n}}, -\rho)$, we get two such intersections. This problem arises due to the ambiguity in the normal direction, and can be solved by considering the normal directions from a half unit sphere ($n_z > 0$ or $n_z < 0$).

In practice, due to efficiency reasons, the Hough space does not consist of parametric shapes whose intersection points are calculated. Instead, a discretized version of the parameter space is used. In Figure 3.2 we show three 2D slices through the discretized 3D Hough space for a given input point cloud coming from a plane. The three slices correspond to three different $\rho$ values. As we approach the correct value of $\rho$ ( the slice in the middle) we get a high accumulator value giving us all three parameters of the plane i.e. $\theta$, $\phi$, and $\rho$. As we use half sphere (corresponding to $n_z > 0$) the problem from normal ambiguity, mentioned above, does not arise.

## 3.3 Cylinder detection

Cylinders are one of the most frequently used primitives for industrial design. This is especially true for processing industries like petrochemical plants, refineries, nuclear plants etc. As a result automatic and robust methods for their detection and fitting are essential for the 3D reconstruction of such sites.

Various methods have been proposed in the literature to fit cylinders to point clouds (Chaperon and Goulette, 2001; Fischler and Bolles, 1981; Lukács et al., 1998;

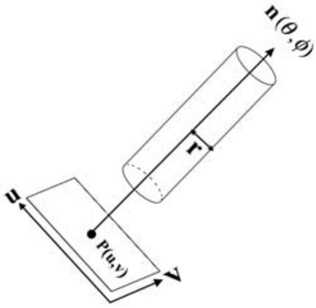


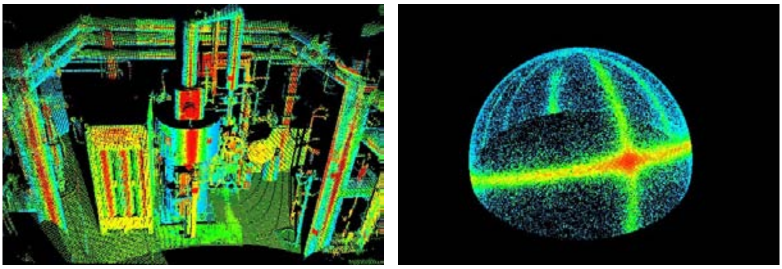
**Figure 3.3:** *The five parameters of a cylinder.* $(\theta, \phi)$ *gives the axis direction in the spherical coordinates,* $r$ *is the radius.* $P(u, v)$ *gives the position in terms of* $\mathbf{u}$ *and* $\mathbf{v}$ *which along with the axial direction* $\mathbf{n} = (\cos\theta\sin\phi \quad \sin\theta\sin\phi \quad \cos\phi)$ *form the local coordinate system of the cylinder.*

Marshall et al., 2001).These methods can be divided into two main categories: those requiring a prior segmentation and those processing raw point clouds without segmentation.

The methods belonging to the first category fit a cylindrical surface to the segmented point cloud. Most of them use non-linear least squares to minimize the orthogonal distance of the points from the fitted cylinder (Lukács et al., 1998; Marshall et al., 2001). These methods assume that the segmentation method used is able to assign correct labels and there are only a few outliers or segmentation errors. As is shown in the comparison of segmentation algorithms for planar surfaces in (Hoover et al., 1996) and for curved surfaces in (Min et al., 2000) these requirements are not met in most cases. The sensitivity of least squares based geometric fitting to outliers is well known (Björck, 1996; Press et al., 1988). Furthermore, non-linear least squares is an iterative process, and to avoid local minima it requires good initial values of the parameters being estimated. In the case of over-segmentation the estimated initial values are poor and the method can get trapped in one of the local minima. In contrast, under-segmentation results in a high percentage of outliers, resulting in an unfaithful reconstruction.

The methods belonging to the second category try to avoid these problems by processing raw point clouds by using robust fitting methods like RANSAC (Bolles and Fischler, 1981; Chaperon and Goulette, 2001; Fischler and Bolles, 1987). For example in (Chaperon and Goulette, 2001) RANSAC is used for cylinder detection and fitting.

The Hough transform based methods have long been used to tackle problems of outliers and multiple instances (Hough, 1962). In noisy and cluttered images they have no parallel in finding lines and curves like circles (Kimme et al., 1975). A major drawback of the Hough transform is its time and space complexity. For geometric fitting problems the space and time complexity can be approximated by $O(s^p)$ and $O(s^{p-1}n)$ respectively, where $n$ is the number of points, $s$ is the number of samples along one Hough dimension, and $p$ is the dimension of the Hough space. Furthermore, in most of the modeling projects employing laser scanning the number of points can be in the order of millions. Due to these reason the use of the Hough transform becomes impractical for the fitting of objects having more than three parameters.



(a) (b)

**Figure 3.4:** *Orientation estimation by the detection of great circles on the Gaussian sphere of the point cloud (a) Input point cloud (b) Gaussian sphere resulting from plotting normals of the point cloud, two great circles corresponding to two cylinder orientations are visible.*

Although a cylinder has five degrees of freedom, different parameterizations to represent it have been proposed in the literature. For example Lukács et al. (1998) uses seven parameters with two constraints. The parameterization we propose is shown in Figure 3.3 and is the best suited for the Hough transform as it uses a minimum number of free parameters with no constraints. Using the previously stated formulas for the space complexity of the Hough transform, for the case of cylinders $p = 5$. This means if we use a small number of 100 samples along each parameter, i.e. $s = 100$, with each cell represented as a 4 byte integer, the required Hough space is 36.3 Giga Bytes (GB). Similarly for a more typical value of $s = 200$, this figure rises to 1164 GB. In practice the value of $s$ may be much larger than 200 resulting in even higher numbers. Thus the use of a 5D Hough space is not practical.

One effective way to reduce the space and time complexity of the Hough transform is to use sequential processing by break down the problem into a set of manageable sub-problems. That is the approach we employ here by dividing the problem of cylinder fitting into two separate steps. The first step uses the Gaussian sphere of the point cloud as its input and consists of a 2D Hough transform. It finds strong hypothesis for the direction of the cylinder axes. In the second step a 3D Hough transform is performed for a few neighboring directions found in step 1, leading to the determination of position and radius of the cylinder. Thus the sequential processing allows us to reduce the effective dimension of the Hough space required for cylinder detection from five to three.

### 3.3.1 Orientation Estimation

The first step in our sequential Hough transform finds strong hypotheses for the cylinder orientation. This orientation estimation is based on the observation that for cylinders the normals make a great circle on the Gaussian sphere (Carmo, 1976). This great circle results from the intersection of the unit sphere with a

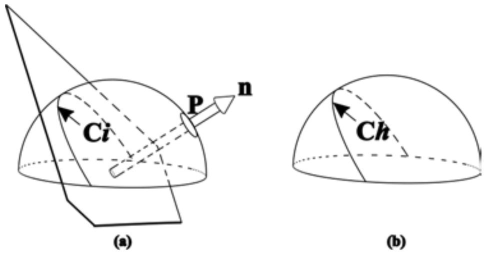


**Figure 3.5:** *How a point* $\mathbf{P}$ *in the input Gaussian sphere votes for a circle in the Hough space (a) Point* $\mathbf{P}$ *on the input Gaussian sphere, votes for a great circle on the Hough Gaussian sphere in (b). This great circle* $\mathbf{C}$ *results from the intersection of the Gaussian sphere with a plane whose normal* $\mathbf{n}$ *equals* $\mathbf{P}$.

plane passing through the origin. The normal of this plane is given by the cylinder axis (Figure 3.4).

The standard Hough transform to find planes in 3D uses a three dimensional Hough space (Section 3.2). In the current case we have two constraints that will enable us to reduce the dimension of the Hough space from three to two. Furthermore, we will exploit these constraints to formulate a rapid update method for the Hough space.

The first constraint comes from the requirement that the plane must pass through the origin, meaning $\rho = 0$. This enables us to remove the third parameter corresponding to the perpendicular distance from the origin, leaving us with a 2D Hough space. Secondly, the plane must intersect the unit sphere, which means that each input point votes for one circular region in the Hough space (Figure 3.5).

Since the Hough transform for this step uses the Gaussian sphere as its input, we need to estimate the normal for each point in the input point cloud. As we do not require the triangulation of the data to be available, the normal estimation methods specific to triangulated data (Petitjean, 2002) cannot be used. Instead, we use a method more suited to unstructured point clouds. This method searches for k nearest neighbors of each point, and then estimates its normal by eigen-analysis of its covariance matrix. For more details see Chapter 2.

As explained above, the constraints in our problem allow us to remove the third parameter, corresponding to the distance of the plane from the origin, from the standard plane-fitting Hough transform. This leaves us with quite a unique situation, because on the input we have the Gaussian sphere of the point cloud, while the Hough space consists of only orientation of the plane normal (representing the great circle and hence the cylinder orientation), which can be interpreted as another Gaussian sphere. To distinguish between these two separate entities we have named the Gaussian sphere resulting from the normals of the input point cloud as *Input Gaussian sphere*, while the one in the Hough space is named the *Hough Gaussian sphere* (Figure 3.6).

Each point in the input Gaussian sphere represents an orientation and it votes for all planes on the Hough Gaussian sphere whose normals are orthogonal to this point. The set of all points in $\mathbb{R}^3$ orthogonal to a given direction forms a plane, but in this case the plane must intersect the Hough Gaussian sphere giving a great circle. Thus each point in the input Gaussian sphere votes for a great circle on the Hough Gaussian sphere. The normal of this great circle is given by the orientation

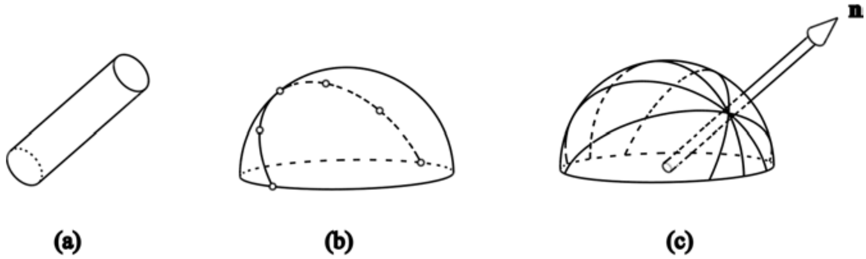


**Figure 3.6:** *Step 1:* Orientation estimation. *(a) Input cylinder. (b) Input Gaussian sphere with the great circle corresponding to the input cylinder. (c) The Hough Gaussian sphere where each point in the input Gaussian sphere votes for a great circle; their intersection estimates the orientation by giving a big value in the Hough space.*

represented by the current point. This voting scheme for a single point is shown in Figure 3.5, while Figure 3.6 shows how individual points belonging to a cylinder (a great circle on the input Gaussian sphere) vote for their corresponding great circles on the Hough Gaussian sphere, whose intersection gives an estimate for the cylinder orientation.

The rapid update method described above requires a parametric equation of the great circle with a given normal expressed in the spherical coordinates. Given this parametric equation each point from the Input Gaussian sphere can directly calculate the respective cells in the Hough Gaussian sphere and vote by incrementing them. The parametric equation for the great circle in the xy-plane with the z-axis as its normal is given by:

$$x = \cos t \quad y = \sin t \quad z = 0 \quad 0 \leq t \leq 2\pi \tag{3.3}$$

To get the points for a great circle with a given normal we need to apply an orthonormal transformation (either a rotation or a reflection) to each point given by Equation 3.3. This transformation matrix must transform the z-axis to the normal of the required circle. Such a reflection matrix can be derived using Householder reflection (Golub and Loan, 1991) as follows:

$$z = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T \tag{3.4}$$

$$n = \begin{pmatrix} \cos\theta\sin\phi & \sin\theta\sin\phi & \cos\phi \end{pmatrix}^T \tag{3.5}$$

$$\mathbf{R} = \mathbf{I} - 2\mathbf{b}\mathbf{b}^{\mathbf{T}} \tag{3.6}$$

Where $\mathbf{b} = \frac{\mathbf{z}-\mathbf{n}}{\|\mathbf{z}-\mathbf{n}\|}$

Using these expressions for a point $\mathbf{P}(\theta,\phi)$ the resulting reflection matrix is given by:

$$\mathbf{R} = \begin{pmatrix} \sin^2\phi - \cos\phi\cos^2\theta & -(1+\cos\phi)\cos\theta\sin\theta & \sin\phi\cos\theta \\ -(1+\cos\phi)\cos\theta\sin\theta & \cos\phi\cos^2\theta - \cos\phi + \cos^2\theta & \sin\phi\sin\theta \\ \sin\phi\cos\theta & \sin\phi\sin\theta & \cos\phi \end{pmatrix} \tag{3.7}$$

Where $(\theta,\phi)$ are the spherical coordinates of the point on the input Gaussian sphere. In the Figure 3.5 and Figure 3.6 we see this voting by the points on the input Gaussian sphere to the cells in the Hough Gaussian sphere.

The algorithm for step 1 of our procedure is as follows:

(a) Calculate the normals for all points in a given point cloud using plane fitting on their k nearest neighbors.

(b) Make a sampled Hough space to represent the Hough Gaussian sphere for the orientation of the cylinder axis using the approximate uniform sampling outlined in appendix A.

(c) For each point in the input data, use the spherical coordinates of its normal calculated in step (a) to derive the matrix $\mathbf{R}$ using Equation 3.7.

(d) Increment the cells in the Hough space given by the rotated parametric form of the circle (Equation 3.3 and 3.7).

(e) Find the points in the Hough Gaussian sphere whose accumulator values are greater than a threshold. These are the hypotheses for the cylinder directions. A filtering step might be necessary to counter the effects of many local maxima.

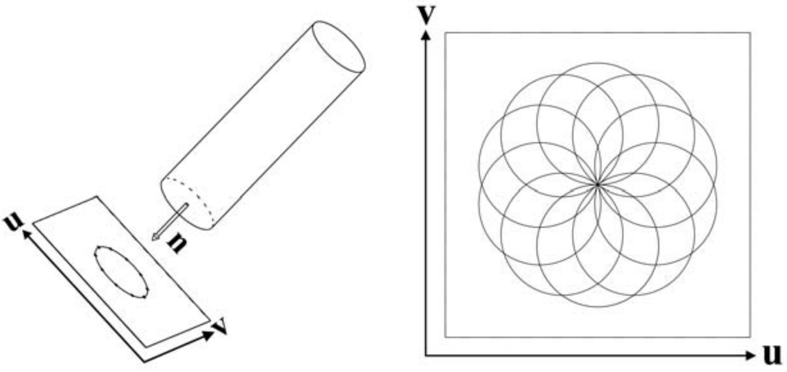
### 3.3.2 Position and Radius Estimation

As explained in Section 3.3.1 a cylinder can be represented by five parameters. Step 1 gives us strong hypotheses of the cylinder orientation. Still there remain three unknown parameters corresponding to the position in the plane perpendicular to the estimated axis and the radius of the cylinder.

We begin step 2 by projecting all the points to the plane perpendicular to the cylinder axis estimated in step 1. For this purpose we need an orthonormal coordinate system with cylinder orientation as one of its axes. There are two possibilities to calculate such a set of orthonormal basis from a given vector, either we can use Gramm-Schmidt orthogonalization (Fraleigh and Beauregard, 1995; Press et al., 1988) or singular value decomposition can be used (Golub and Loan, 1991). Both of these options produce an orthonormal coordinate frame for the projection consisting of three basis vectors, $\begin{pmatrix}\mathbf{u} & \mathbf{v} & \mathbf{n}\end{pmatrix}$ where $\mathbf{n}$ equals the cylinder axis (from step 1), and $\mathbf{u}$ and $\mathbf{v}$ are the other two orthonormal basis vectors.

After the projection to the plane, we proceed to calculate the position and the radius of the cylinder using circle fitting on the projected points. The Hough transform for circle fitting is based on the formulation given by Kimme et al. (1975). For a given radius $\mathbf{r}$, each projected point votes for the bins in a circular region in the Hough space with the current point as their center. If the projected coordinates of a point are given by $(u_p, v_p)$, it votes for the cells in the Hough space given by:

$$\begin{pmatrix} r\cos\omega + u_p & r\sin\omega + v_p \end{pmatrix} \qquad 0 \leq \omega < 2\pi \tag{3.8}$$



(a) (b)

**Figure 3.7:** *Step 2:* position and radius estimation *(a) Points are projected along the estimated orientation or one of its neighbors (b) A 2D slice through the 3D Hough space corresponding to the correct radius. Each projected point votes for a circle that represents the locus of the possible circle centers. For the correct radius all these circles intersect, thus giving the highest count in the Hough space.*

The peak in the Hough space estimates the radius of the cylinder directly. However, the position is still in the projection coordinate system calculated above. It must be transformed back to the world coordinate system to get a 3D point on the axis of the cylinder. This coordinate transformation is given by the following matrix $\mathbf{T}$:

$$\mathbf{T} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix} \tag{3.9}$$

The above described process is performed for each orientation found in step 1 along with some of its neighbors on the uniform sampling of the Gaussian sphere (See appendix A). The inclusion of these neighbors results in the refinement of the orientation estimate in addition to the determination of the position and radius of the cylinder.

The algorithm for step 2 of our procedure can be summarized as follows:

(a) For each orientation found in step 1, find its $N$ nearest neighbors. For this neighbor search use approximate uniform sampling (see appendix A).

(b) For each orientation $\mathbf{n}$ derive an orthonormal coordinate system consisting of three basis vectors given by $\begin{pmatrix}\mathbf{u} & \mathbf{v} & \mathbf{n}\end{pmatrix}$ using Gramm-Schmidt orthogonalization or singular value decomposition. Project all the points to this new coordinate system.

(c) For each value of radius $r$ in a user-specified radius range, increment cells in the Hough space given by Equation 3.8 for each projected point given by $\begin{pmatrix}u_p & v_p\end{pmatrix}$ .

(d) Find the peak in the 3D Hough space. This gives the cylinder radius directly. Transform the position by the transformation matrix $\mathbf{T}$ (Equation 3.9) to the world coordinate system

(e) Remove the points corresponding to the found cylinder from the data. If enough points remain, proceed with the step 1 for the detection of next cylinder.

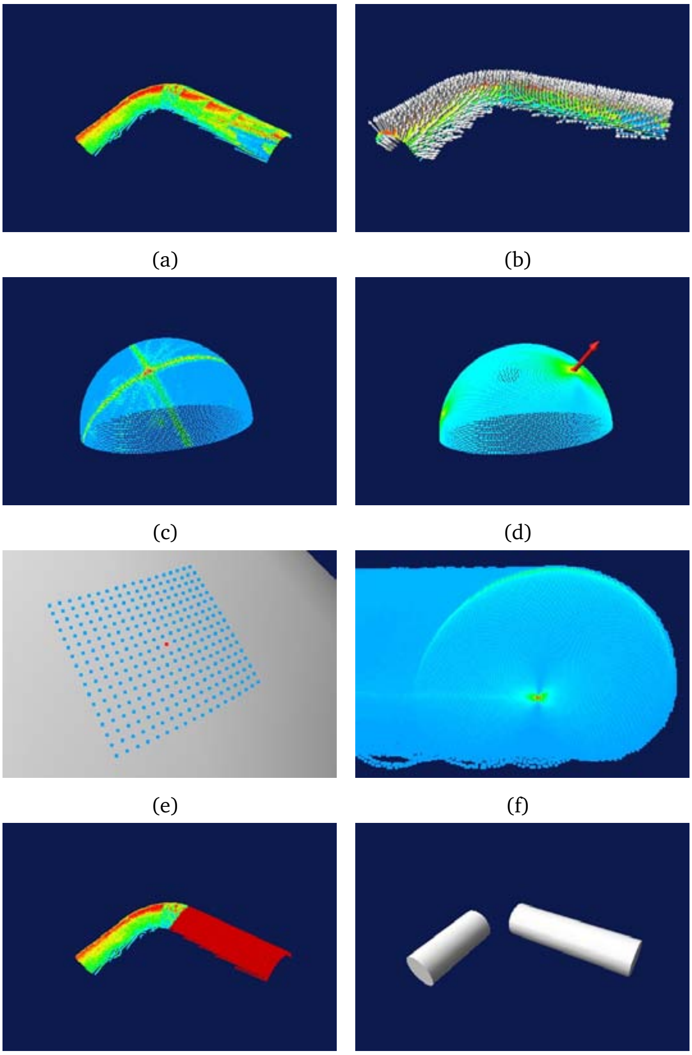
Figure 3.7(a) shows the results of the projection, while Figure 3.7(b) shows a 2D slice through the three-dimensional Hough space corresponding to the correct radius.

### 3.3.3 Example

To explain the above described steps, in Figure 3.8 we show the step-by-step process for a point cloud captured from an L-junction. In 3.8(a) we show the original point cloud, 3.8(b) shows the estimated normals as arrows. 3.8(c) and 3.8(d) illustrates the step 1 of orientation estimation, with 3.8(c) showing the input Gaussian sphere with two great circles corresponding to the two different cylinder orientations. In Figure 3.8(d) the Hough Gaussian sphere has two distinct peaks detecting the two strong hypotheses for cylinder orientations. In Figure 3.8(e) we illustrate the process of neighboring orientation selection on uniformly sampled Gaussian sphere, which is necessary for the refinement of the orientation estimation coming from step 1. Figure 3.8(f) shows the 2D slice through the 3D Hough space for position and radius estimation (step 2). The slice corresponds to the correct radius where the peak gives the position of the cylinder. 3.8(h) shows the final result, where both cylinders are successfully detected. For the results shown in Figure 3.8 the Hough space for step 1 consisted of $250,000$ cells, whereas Step 2 used a 3D Hough space having $512 \times 512 \times 100 \cong 26 \times 10^6$ cells. In contrast a straight forward 5D Hough space for equivalent results would have required $512 \times 512 \times 100 \times 512 \times 512 \cong 7 \times 10^{12}$ cells, which even for today's ever expanding standards is quite impractical. However, the presented sequential Hough transform reduces the space complexity, making the problem manageable.

### 3.3.4 Results of cylinder detection

Figure 3.9 shows the results of the presented algorithm on two data sets from two industrial sites. During their processing it was found that during the step 1 of orientation estimation the large clusters on the input Gaussian sphere, resulting from big planar areas, interfered with the step 1 of cylinder axis estimation. To resolve this problem, the data sets were preprocessed with a plane-fitting Hough transform (Section 3.2 to remove planar areas. A technique for simultaneous detection of planes and cylinders based on hypothesis verification is presented in the next section.



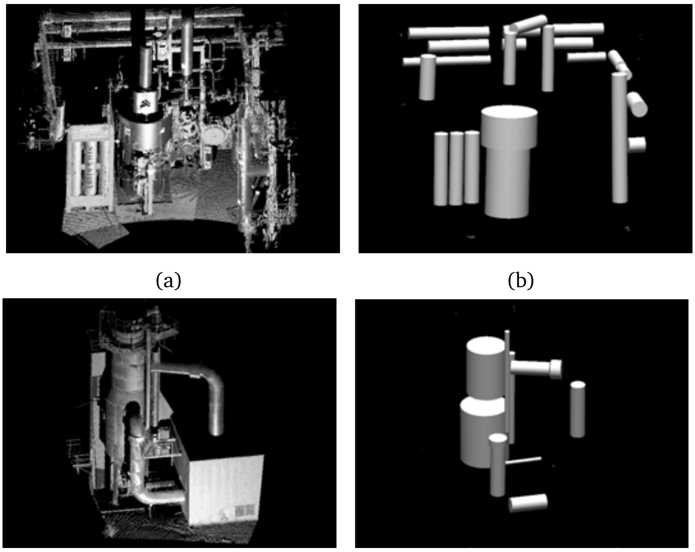
(a) (b)

(c) (d)

(e) (f)

(g) (h)

**Figure 3.8:** *Step-by-step processing of the point cloud of an L-junction (a) Input data (b) Estimated normals, notice the ambiguity in outward pointing direction, justifying the use of the half sphere (c) Input Gaussian sphere. There are two great circles one corresponding to each cylinder direction (d) Hough Gaussian sphere. We get two distinct peaks; the highest peak carrying the red arrow is for the longer cylinder (e) Selection of the neighboring orientations using approximate uniform sampling of the unit sphere. The orientation predicted by Step 1 is in the center. (f) Step 2 of the position and radius estimation shown for the correct orientation and radius (a 2D slice through 3D Hough space). The peak gives the position of the cylinder. (g) The points belonging to one cylinder selected automatically based on distance threshold and normal similarity (h) Final result, both cylinders have been automatically found.*


(a) (b)

(c) (d)

**Figure 3.9:** *Results of the Hough transform for cylinder detection. (a and c) Input point cloud (b and d) Results of cylinder detection.*

Figure 3.9(a) shows point cloud 1, while Figure 3.9(b) shows the detected cylinders. The sequential Hough transform has been able to detect cylinders not only in different orientation, but also multiple radii along one orientation have been successfully found.

This success can be explained by the sequential nature of our algorithm, because for each orientation found in step 1, we look for positions and radii within a user-specified bound in step 2. The peak corresponds to one cylinder. Next, we select the points belonging to this cylinder, based on a distance and normal similarity threshold. The selected points are removed from the input point cloud, and if sufficient points remain we proceed with the next iteration.

Figure 3.9(c) and Figure 3.9(d) show another dataset from an industrial plant along with the found cylinders. As for this dataset wider radius bounds were specified, cylinders of both small and big radii were successfully detected.

## 3.4 Ambiguity resolution by hypothesis testing

The Hough transform is an automatic model detection technique through voting in a discretized parameter space which is robust to outliers and can also detect multiple instances of a given type of object. A big value in the Hough space indicates that there is a high probability that one or more instances of the object represented by this cell in the parameter space does exist in the input data.

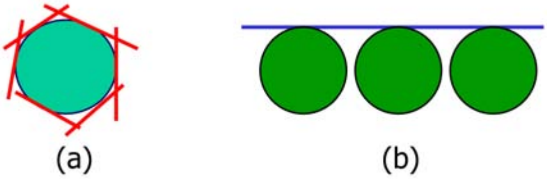


**Figure 3.10:** *Ambiguity between the cylinder and the plane (a) Multiple planar patches can be fitted to a cylinder (b) multiple cylinders fitted to a plane*

However, without verification of this hypothesis, there is no guarantee that the detection is correct and valid. Consequently, the Hough transform must be followed by a verification and validation stage.

The simplest verification scheme is to select the points within a certain distance $d_{th}$ from the detected object, and if the number of the selected points exceeds a given threshold $n_{th}$, the hypothesis is said to be verified. This scheme works quite well in 2D problem and is used almost exclusively for line and circle detection in images.

In 3D the data has more degrees of freedom, and as a next refinement we can add normal similarity to distance thresholding. For a given point cloud the normals are estimated using the techniques from Chapter 2. Only those points whose normal makes an angle less than $\theta_{th}$ with the predicted ideal normal of the detected object are deemed suitable for verification.

An additional problem arises when two or more different types of objects are being simultaneously detected, but their domains are not mutually exclusive. This results in an inherent ambiguity which must be resolved by providing some extra information in terms of distinguishing characteristics of the ambiguous models. Such an ambiguity arises for example when planes and cylinders are being simultaneously detected. A cylinder can be locally represented by a planar strip (Figure 3.10(a)). Similarly a plane can be represented by a cylinder of infinitely big radius. Just putting a threshold on the radius of acceptable cylinders does not solve the problem, because a big plane is now divided into a series of cylinders each of which fits a long sub-strip of the plane(Figure 3.10(b)). All these solutions are mathematically valid, but not desirable, as they do not match the response of a human operator.

This implies that just distance and normal based hypothesis verification does not possess enough discriminating power to resolve this ambiguity. Such a simple verification scheme will either split all cylinders into narrow planar strips or all planes will be divided into a series of cylinders. What is needed is a set of constraints expressed in terms of some quantitative measure that summarizes the user's notion of what is an acceptable cylinder or an acceptable plane. Next we present such measures for the ambiguity resolution between planes and cylinders. We call them *Planarity* and *Cylindricality* test.

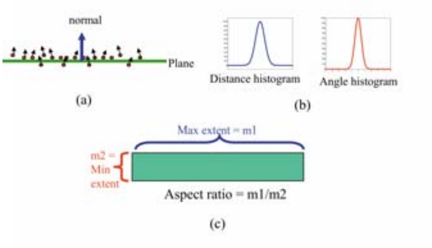


**Figure 3.11:** *Planarity test (a) Distance of the points from the detected plane and the local point normals vs ideal plane normal (b) histogram of the distances from the plane and histogram of the angle between point normals and the normal of the plane. For a planar region both of these histograms should look like a very peaked distribution (c) Aspect ratio can help avoid fitting of long strips of planes to cylinders.*

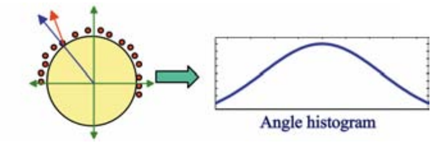


**Figure 3.12:** *Cylindricality test. For accepting a cylinder a histogram of the angular coverage of points is used. For example 25% or higher coverage could be required.*

### 3.4.1 Planarity test

The intuitive notion of a plane that distinguishes it from a cylinder can be described by a combination of following measures (Figure 3.11).

1. The histogram of the distances of the points from the detected plane (or residuals of plane fitting) should look like a peaked distribution (Figure 3.11(a)). One measure of peakedness is the sum of the squares of the histogram entries divided by the squares of the number of points. This measure can have a maximum value of 1. Another measure of peakedness, often used in statistics, is *kurtosis* which is based on the 4-th moment of the histogram (Weisstein, 2005).

2. The histogram of the angles of the points with the plane normals should be peaked as well (Figure 3.11(a)). For measuring the peakedness one of the techniques suggested above can be used.

3. To avoid the splitting of cylinders into multiple thin planar slices, we require the aspect ratio of the region to be close to 1 (Figure 3.11(b)). The aspect ratio is defined as the ratio of the maximum to minimum extent of the points, after they have been projected to the detected plane. For this projection techniques similar to Section 3.3.2 can be used. A very high aspect ratio means a very thin slice, which might be coming from a cylinder, and can thus be rejected.

Note that this planarity measure is calculated after the initial selection of the points, based on the distance and the normal similarity threshold described above.

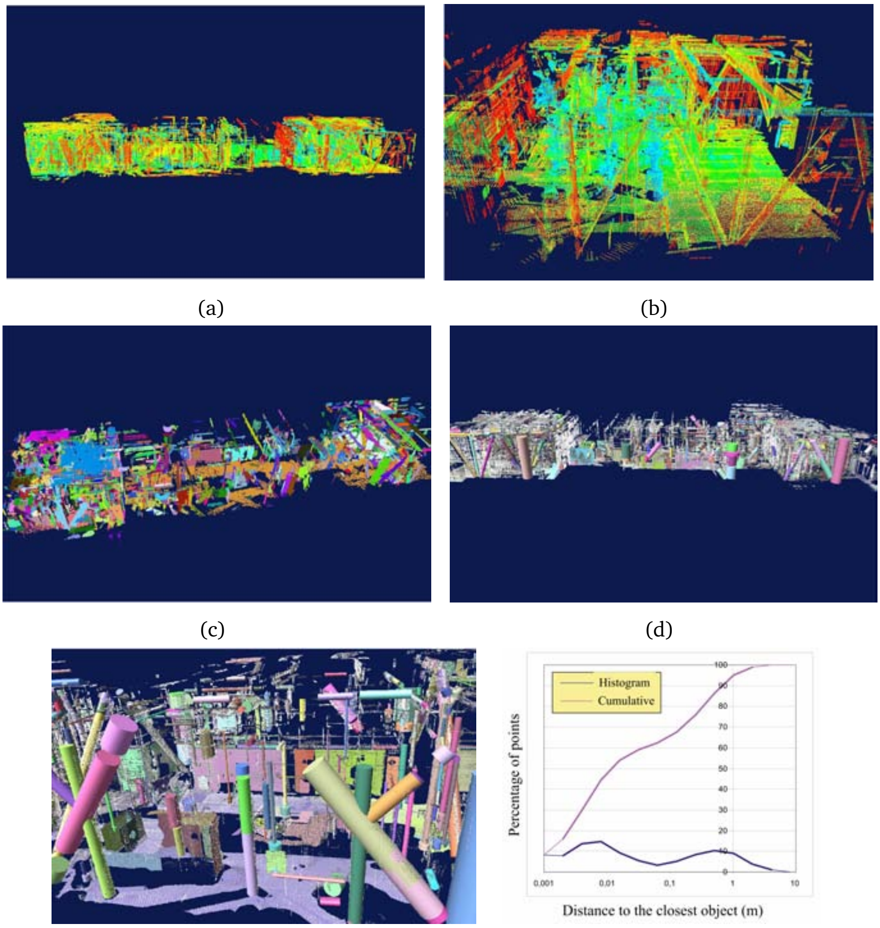
### 3.4.2 Cylindricality test

For accepting a cylinder we use the measure consisting of following quantities.

1. We assume that a significant portion of the cylinder is visible to the scanner. A typical significant portion can be 25% or more of the cylindrical surface. This notion can be quantified by counting the number of points whose normals make a certain angle with a reference direction. We can also put it in terms of the great circle, which if divided into $n$ bins, should result in at least 25% the bins containing points more than a given threshold (Figure 3.12). This test takes care of a cylinder being partially fitted to a plane, as there the coverage can never be more than 5-10% (provided the radius limits are realistic).
2. The length of the cylinder should be greater than a given threshold. It is not desirable to have a lot of tiny cylinders. The length is measured by projecting the points along the axial direction and creating a its discretized image. All the bins in this discretized image with a count greater than a threshold are kept, and the linear segments are found through connected component analysis. All linear segments (representing connected points on the cylinder) having a length greater than a threshold $l_{th}$ are kept.
3. There should be a minimum and maximum limit on the radius of cylinders. These limits will speed up the second step of cylinder detection (Section 3.3.2). Additionally, in combination with measure 1 above it solves the problem of infinite cylinders or multiple stacked cylinders being fitted to planar surfaces.

## 3.5 Results

The simultaneous method of plane and cylinder detection through the presented Hough transforms combined with the stage of ambiguity resolution was applied to a data set from a petrochemical industrial plant. The point cloud consisted of 20 million points, which were first segmented using the methods presented in Chapter 2. Each segment was then processed through the presented Hough transforms for plane and cylinder detection. As the segmentation based on the smoothness constraint avoids over-segmentation, processing each segment separately provides a better result by localizing the parameter space for the Hough transform. For ambiguity resolution the following parameter settings were used for planarity and cylindricality tests. The distance threshold $d_{th}$ was set at 20mm, the angle threshold $\theta_{th}$ at $15^\circ$ and the minimum number of points threshold $n_t h$ at 2000. For a cylinder a minimum surface coverage by measured points of 25% or $90^\circ$ was required, the acceptable radius limits were $50mm < r_{cyl} < 1m$ and the minimum cylinder length was set at 200mm. For planarity test the maximum acceptable aspect ratio was set at 5 and the histogram peakedness measures was required to be more than 0.75. This resulted in a fully automatic detection of 946 planar and 1392 cylindrical patches. The results are shown in Figure 3.13.



(a) (b)

(c) (d)

(e) (f)

**Figure 3.13:** *Results on the point cloud from a petrochemical plant (a-b) input point cloud (c) results of segmentation (d-e) detected planes and cylinders (f) histogram of the distance to the closest model*

To analyze the results of this automatic detection the histogram of the distance of all points in the input point cloud to the closest model detected through the Hough transform is shown in Figure 3.13(f). We see the first peak at 8mm, which is close to the point accuracy of Cyrax scanner (6mm) which was used for the data acquisition. From the plot we see that more than 50% of the points have a distance of less than 1cm to the closest modeled surface. The distances of 2cm to 10cm are caused by un-modeled surfaces like elbows, screws, valves and other structure present on the site, which cannot be represented by planes and cylinders. About 55-60 % of the points have been used for the automatically detected objects.

## 3.6 Conclusions

We have presented extensions of the Hough transform for the automatic detection of planes and cylinders in the point clouds of industrial facilities. Although the extension for plane detection is quite straight-forward, the space-complexity for cylinder detection arising from its 5 parameters, required a two-step approach. This enabled us to employ a combination of a 2D and a 3D Hough transform, making the problem manageable. The results showed this sequential breakdown still maintains the advantage of robustness against outliers and multiple instances that necessitated the use of the Hough transform in the first place.

As the domains of the planar and the cylindrical models overlap, the resulting ambiguity was solved through imposition of extra constraints. These constraints were summarized in terms of a planarity and a cylindricality test that allowed us to select the appropriate model in the same manner as a human operator would do. As the planes and the cylinders are two of most commonly found objects on industrial installations, their automatic detection can decrease the manual effort required for reverse engineering. We presented results on some data-sets from industrial installations, which show the strength and success of the presented approach.

# Chapter 4

# Model fitting to point clouds

> Pursuit of the approximate can conclude. Not so pursuit of the absolute.
>
> Mason Cooley (b. 1927)

> Nature knows nothing but solid bodies; your science deals only with combinations of surfaces.
>
> Honor De Balzac (1799-1850)

## 4.1 Introduction

In this chapter we address the problem of fitting a given model to a set of 3D points. As argued in Chapter 1, modeling is a necessary process for converting the point cloud to a useful CAD representation. It provides a complete picture of the as-built situation by filling the gaps coming from occlusions, by averaging the effects of noise and by providing the quality measures about the final results.

In the previous chapter we presented methods based on the Hough transform for the automatic detection of planes and cylinders in a segmented point cloud. The Hough transform must be followed by a fitting stage because the quality of the parameters estimated there depends on the granularity of the Hough space. By fitting a model to the points selected through the Hough transform, the side-effects arising from the quantization of the Hough space are eliminated. Furthermore, the parameters of other categories of simple objects like spheres, cones and tori, which have not been automatically detected, must be estimated through model fitting. The estimation process also provides a tool to check and measure the deformations in the model by statistical analysis of resulting residuals. For this purpose we present methods for fitting simple objects through least squares.

In the traditional modeling pipeline, the fitting of the simple objects is followed by a manual editing stage to produce a final Boundary representation (B-rep) model. This is a labor-intensive process and the quality of the results are usually poor as they are solely based on human interpretation without using either the acquired data or the a priori information. We present an automatic method for this conversion by fitting a complex model represented as Constructive Solid Geometry (CSG) tree to the selected subset of points. CSG is frequently used as a representation scheme for industrial design in many sectors. For example in petrochemical industries most of the CAD packages use object libraries based on CSG (AVEVA PDMS, 2005; Intergraph PDS, 2005). As the initial design of many sites is carried out using these CAD libraries, the use of CSG during reconstruction comes naturally and is justified. Moreover, CSG encodes the a priori shape knowledge in the form of geometric constraints, whose enforcement further reduces the degrees of freedom and thus leads to a better estimation of the parameters of the model.

The rest of the chapter is organized as follows. In Section 4.2 we give a mathematical definition of the general model fitting problem and present the details of its nonlinear least squares counterpart, which is used exclusively in the next sections. Section 4.3 presents methods for fitting of simple objects like planes, spheres, cylinders, cones, and tori to the point cloud. The techniques for getting good approximate values, which are essential for the convergence of the iterative nonlinear least squares methods, are also presented. In Section 4.4 we present methods for fitting of models represented as CSG trees to given point clouds. Section 4.5 contains some results and comparisons, and finally we conclude in Section 4.6.

## 4.2 Problem definition

Model fitting can be posed as an optimization problem, where we search for those parameters of a given type of model which lead to the best agreement between the selected points and the resultant model. This degree of agreement for one point can be assessed by its distance from the model surface, while for the point cloud it can be measured by the sum of a function of the distances for all points. A lower value of this sum indicates a higher degree of match and vice versa. Two distance measures typically used are the algebraic and the geometric distance.

Mathematically the model fitting problem can be formulated as follows:

$$\min_{\tau_1,\tau_2,\cdots\tau_m} \sum_{i=1}^{N} \mathcal{F}(\Psi[p_i, \Gamma(\tau_1, \tau_2, \ldots, \tau_m)]) \tag{4.1}$$

In Equation 4.1 $\Gamma(\tau_1, \tau_2, \ldots, \tau_m)$ is the model being fitted to the given point cloud consisting of $n$ points $p_1, p_2, \cdots, p_n$. The model is parametrized by $m$ shape and pose parameters $\tau_1, \tau_2, \ldots, \tau_m$. $\Psi$ is a function giving the distance (algebraic, orthogonal or some other distance measure) of the $i-th$ point $p_i$ from the model, and $\mathcal{F}$ is a function of this distance ($(.)^2$ for least squares, $|.|$ for M-estimator etc). $\mathcal{F}$

should be even i.e. $\mathcal{F}(-x) = \mathcal{F}(x)$, so that the negative and the positive distances do not cancel each other.

In this chapter we will look at the problem of fitting different types of surfaces ($\Gamma$ in Equation 4.1) like planes, spheres, cylinders, cones, tori and CSG models to a given point cloud. We will use the orthogonal distance for $\Psi$ as the comparison with the algebraic distance in Section 4.2.2 will show the orthogonal distance to give a superior performance. For $\mathcal{F}$ we will use the square of orthogonal distance i.e. $(.)^2$, converting the above problem to the following general least squares problem:

$$\min_{\tau_1,\tau_2,\cdots\tau_m} \sum_{i=1}^{N} \Psi^2[p_i, \Gamma(\tau_1, \tau_2, \ldots, \tau_m)] \tag{4.2}$$

### 4.2.1 Fitting method

The problem in Equation 4.2 can be solved by one of the standard non-linear least squares methods. We decided to use Levenberg-Marquardt method as it provides better convergence properties by adaptively combining the step given by Newton's method and that of the steepest descent (Bertsekas, 1995; Björck, 1996; Dennis and Schnabel, 1996; Press et al., 1988).

Starting from an initial estimate of the model parameters $\Gamma_0$, at each iteration we get an adjustment $\Delta\Gamma$ given by:

$$\begin{aligned} \Delta\Gamma &= (\mathbf{J}^T\mathbf{W}\mathbf{J} + \lambda\mathbf{I})^{-1}(\mathbf{J}^T\mathbf{W}\mathbf{D}) && (4.3) \\ \Gamma_1 &= \Gamma_0 - \Delta\Gamma && (4.4) \end{aligned}$$

Here $\mathbf{J}$ is the Jacobian matrix and $\mathbf{D}$ is the distance vector given by:

$$\begin{aligned} \mathbf{J}_{ik} &= \frac{\partial\Psi_i}{\partial\tau_k} = \frac{\partial\Psi(p_i, \Gamma_0)}{\partial\tau_k} && (4.5) \\ \mathbf{D}_i &= \Psi(p_i, \Gamma_0) && (4.6) \end{aligned}$$

$\mathbf{W}$ is the weight matrix giving the quality of each point measurement and $\lambda$ is the Levenberg-Marquardt parameter. For $\lambda = 0$ Newton step is taken. As $\lambda \to \infty$ the iteration leans towards steepest descent step. A typical procedure for the adaptive selection of $\lambda$ is given by Press et al. (1988, chap. 15.5).

The covariance matrix $\mathbf{C}$ of the estimated parameters can be computed by:

$$\mathbf{C} = (\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1} \tag{4.7}$$

For more details see Bertsekas (1995); Björck (1996); Dennis and Schnabel (1996).

### 4.2.2 Different distance measures

In Equation 4.1 and 4.2 different distance measures $\Psi$ between a 3D point and the model being fitted can be used. Two such frequently used measures are the *Algebraic distance* and the *Geometric* or *Orthogonal distance*.

**Algebraic distance** This distance measure is defined only for those surfaces which can be expressed as an implicit function (Chapter 1). The zero set of the implicit function gives the model surface i.e. $f(x, y, z) = 0$. For any point on the model surface the implicit function equals zero. For any point $\mathbf{p}_o = \begin{pmatrix} p_{ox} & p_{oy} & p_{o_z} \end{pmatrix}$ off the surface the value of the implicit function $f(p_{ox}, p_{oy}, p_{oz})$ is related to the distance, though this relation depends on the type of the surface. For example in the case of a sphere of radius $r$ and centered at $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}$, the algebraic distance which equals the value of the implicit function is given by $d_{algebraic} = (p_{ox} - c_x)^2 + (p_{oy} - c_y)^2 + (p_{oz} - c_z)^2 - r^2$ while the orthogonal or geometric distance of the point from this sphere would be $d_{geometric} = \sqrt{(p_{ox} - c_x)^2 + (p_{oy} - c_y)^2 + (p_{oz} - c_z)^2} - r$. Which means that there is a nonlinear relationship between the algebraic and the geometric distance and it can be said that the algebraic distance for the sphere approximately equals the square of the geometric distance. This square relationship results in a higher sensitivity to the outliers and noise for the algebraic distance compared to the geometric distance.

There are two attractive features of the algebraic distance. Firstly, its calculation is very simple provided an implicit functional form for the given model exists. Secondly, the implicit function formulation results in a linear least squares problem which has a non-iterative closed-form solution. As a result the algebraic distance based fitting can be used as a quick way to calculate approximate values.

**Geometric distance** The smallest distance of a point from the model surface, so that the local surface normal and the line joining the closest point to the given point are collinear, is called the Geometric or the Orthogonal distance. In most cases the calculation of the orthogonal distance is more complicated than the corresponding calculation of the algebraic distance. In contrast to the algebraic distance which is defined only for those surfaces which can be expressed as an implicit function, the geometric distance is defined for all surfaces. When using the least squares estimation, the geometric distance always results in a nonlinear (some times constrained nonlinear) least squares fitting problem and thus necessitates the use of an iterative method for its solution. In general the use of the geometric distance results in a better solution compared to the algebraic distance. This is especially true in the presence of noise and small outliers.

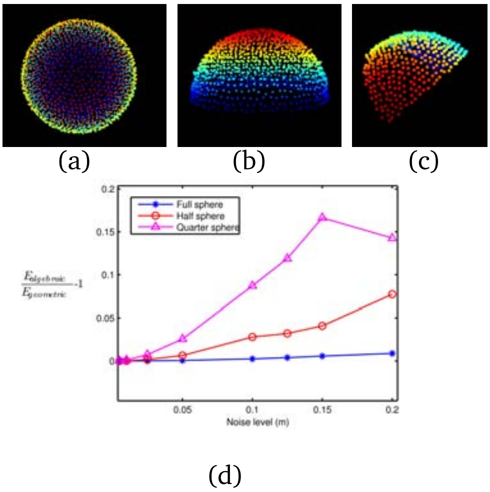


**Figure 4.1:** *The algebraic vs the geometric distance fitting for a sphere (a) Full sphere (b) Half sphere (c) Quarter sphere (d) The noise level vs the ratio of the mean squared error of the least squares fitting based on the algebraic distance to the one based on the geometric distance* $\frac{\epsilon_{algebraic}}{\epsilon_{geometric}} - 1$.

| Noise level | $\frac{\epsilon_{algebraic}}{\epsilon_{geometric}} - 1$ | | |
|---|---|---|---|
| (m) | Full sphere | Half sphere | Quarter sphere |
| 0.001 | 2.62e-7 | 1.05e-6 | 7.2e-4 |
| 0.003 | 1.52e-6 | 5.42e-6 | 1.27e-5 |
| 0.005 | 6.78e-6 | 7.94e-5 | 2.2e-4 |
| 0.010 | 2.53e-5 | 2.54e-4 | 0.0012 |
| 0.025 | 1.62e-4 | 1.75e-3 | 7.36e-3 |
| 0.050 | 6.39e-3 | 6.48e-3 | 0.0255 |
| 0.100 | 2.51e-3 | 0.027 | 0.087 |
| 0.125 | 3.9e-3 | 0.032 | 0.119 |
| 0.150 | 5.7e-3 | 0.041 | 0.166 |
| 0.200 | 8.9e-3 | 0.077 | 0.143 |

**Table 4.1:** *Comparison of fitting based on the algebraic and the geometric distance for a sphere with different point coverages and noise levels.*

### Comparison of distance measures for sphere fitting

To compare the performance of the geometric distance with the algebraic distance we generated points from a unit sphere with different amounts of Gaussian noise. Three different versions of the sphere were used, a full sphere, a half sphere and a quarter sphere to test the sensitivity of the method to partial coverage (Figure 4.1). The results of $\frac{\epsilon_{algebraic}}{\epsilon_{geometric}} - 1$, are given in Table 4.1 and plotted in Figure 4.1(d). $\epsilon$ measures the mean squared error after least squares fitting. From Table 4.1 we see that compared to the geometric distance, the algebraic distance based fitting always results in a higher mean squared error. The relative performance of the algebraic distance deteriorates as more noise is added. Moreover, the error of the algebraic distance based fitting is strongly influenced by the area coverage as the results for the full sphere are better than those for the half sphere. The results for the quarter sphere are the worst. The final dip in the graph for the quater sphere is due to some numerical effect that leads to a fall in the relative error.

This comparison shows that in general the use of the geometric distance results in a better estimation of the model, which is less sensitive to noise and area coverage. Nevertheless, because of the closed form solution the algebraic distance based fitting can be used for getting good approximate values. For more details about the comparison of the accuracy and speed of algebraic vs. the orthogonal distance for different types of surfaces see Ahn et al. (2002); Helfrich and Zwick (1993); Tucker and Kurfess (2003)

In the following sections we will address the problem of model fitting through the least squares estimation based on the geometric distance. For getting the approximate values the algebraic distance based quadric fitting will be used.

## 4.3 Fitting of simple geometric objects

In this section we will give methods for fitting planes, spheres, cylinders, cones and tori to a set of 3D points. We will start with quadric fitting based on the algebraic distance, which will be used for calculating good approximate values for the sphere, the cylinder and the cone.

### 4.3.1 Quadric fitting for approximate values

A quadric is a second degree surface expressed by the following implicit equation:

$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0 \tag{4.8}$$

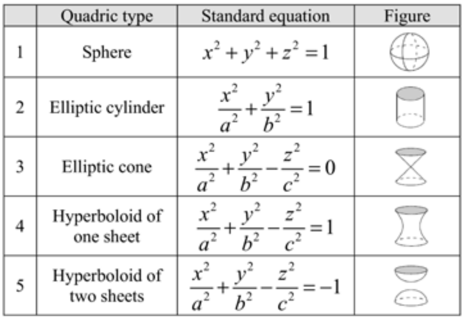
As for fitting any other model, here also we have to choose between the geometric and the algebraic distance (Section 4.2.2). A method for the calculation of the geometric distance of a point from a quadric surface is given by Eberly (2001), which requires finding the roots of a 6-th degree polynomial. As we are interested in an approximate solution, the algebraic distance can be safely used. Next we will present the linear least squares problem which arises for quadric fitting based on the algebraic distance.

From Equation 4.8 we see that a quadric has only 9 degrees of freedom but we have ten coefficients. This over-parametrization can be taken care of by putting a constraint on the parameters of the quadric. If we choose the following constraint:

$$a^2 + b^2 + c^2 + f^2 + g^2 + h^2 + p^2 + q^2 + r^2 + d^2 = 1 \tag{4.9}$$

the resulting problem can be solved through the use of Lagrange multipliers. By following a derivation similar to the one given for plane fitting in Chapter 2, the problem can be reduced to an eigenvalue problem.

If we define the following matrices for a point cloud having $n$ points, where $n >= 9$.



| | Quadric type | Standard equation | Figure |
|---|---|---|---|
| 1 | Sphere | $x^2+y^2+z^2=1$ | |
| 2 | Elliptic cylinder | $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$ | |
| 3 | Elliptic cone | $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=0$ | |
| 4 | Hyperboloid of one sheet | $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1$ | |
| 5 | Hyperboloid of two sheets | $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=-1$ | |

**Table 4.2:** *The subset of the quadric surfaces used for getting approximate values for the sphere (row 1), the cylinder (row 2), and the cone (rows 3-5) and their standard equations. The quadric is estimated through linear least squares based on the algebraic distance.*

:

$$
\begin{aligned}
\mathbf{A} &= \begin{pmatrix} x_1^2 & y_1^2 & z_1^2 & 2y_1z_1 & 2z_1x_1 & 2x_1y_1 & 2x_1 & 2y_1 & 2z_1 & 1 \\ x_2^2 & y_2^2 & z_2^2 & 2y_2z_2 & 2z_2x_2 & 2x_2y_2 & 2x_2 & 2y_2 & 2z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 1 \\ x_n^2 & y_n^2 & z_n^2 & 2y_nz_n & 2z_nx_n & 2x_ny_n & 2x_n & 2y_n & 2z_n & 1 \end{pmatrix} && (4.10) \\
\mathbf{n} &= \begin{pmatrix} a & b & c & f & g & h & p & q & r & d \end{pmatrix}^T && (4.11) \\
\mathbf{M} &= \mathbf{A}^T A = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T && (4.12)
\end{aligned}
$$

It can be shown that the constrained least squares solution for $\mathbf{n}$ is given by that eigenvector of $\mathbf{M}$ ($i^{th}$ column of matrix $\mathbf{E}$) which corresponds to its minimum eigenvalue ($i^{th}$ diagonal element $\lambda_i$ of matrix $\mathbf{\Lambda}$ above).

The quadric given by the vector of parameters $\mathbf{n}$ estimated above is not yet in the standard form and contains a residual rotation and shift. Next we will present a method for their calculation so that the estimated quadric can be transformed to its standard equation (Table 4.2).

### Canonical analysis of the quadratic form

Canonical analysis is a method to reduce a given quadratic form to one of the standard forms (Table 4.2) by an appropriate rotation and translation of variables. The rotation removes the cross product terms and the translation takes care of the linear terms (Box and Draper, 1986, chap. 10.3);(Borg and Bielajew, 1995);(Fraleigh and Beauregard, 1995, chap. 8).

As we are going to use canonical analysis only on the quadric surfaces, henceforth we will derive the method only for that case. Equation 4.8 can be written in the matrix form as follows:

$$\mathbf{v^T Q v} + \mathbf{v^T b} + d \tag{4.13}$$

where $\mathbf{Q} = \begin{pmatrix} a & h & g \\ h & b & f \\ g & f & c \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} 2p \\ 2q \\ 2r \end{pmatrix}$ and $\mathbf{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

Let $\lambda_1, \lambda_2, \lambda_3$ be the eigen values of the symmetric matrix $\mathbf{Q}$ and $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ be the corresponding eigen vectors. We require that all eigen vectors are of a unit length i.e. $\|\mathbf{e}_i\| = 1$. Let $\mathbf{E}$ be a $3 \times 3$ orthogonal matrix whose columns consists of these eigen vectors, and $\mathbf{\Lambda}$ be a diagonal matrix with corresponding eigen values as diagonal entries, then by definition we have

$$\mathbf{QE} = \mathbf{E\Lambda} \tag{4.14}$$

Pre-multiplying both sides by $\mathbf{E}^T$ and using the fact $\mathbf{E}^{-1} = \mathbf{E}^T$ gives

$$\mathbf{E}^T\mathbf{QE} = \mathbf{\Lambda} \tag{4.15}$$

Using $\mathbf{EE}^T = \mathbf{I}$ Equation 4.13 can be written as:

$$(\mathbf{v}^T\mathbf{E})\mathbf{E}^T\mathbf{QE}(\mathbf{E}^T\mathbf{v}) + (\mathbf{v}^T\mathbf{E})(\mathbf{E}^T\mathbf{b}) \tag{4.16}$$

Lets now define $\mathbf{v}_a = \mathbf{E}^T\mathbf{v}$ and $\mathbf{b}_a = \mathbf{E}^T\mathbf{b}$. Substituting these in Equation 4.16 we get

$$\mathbf{v}_a^T\mathbf{\Lambda}\mathbf{v}_a + \mathbf{v}_a^T\mathbf{b}_a + d \tag{4.17}$$

As the matrix $\mathbf{\Lambda}$ in the above expression is diagonal the cross terms are removed and we are left with an equation having only linear terms and a constant. The linear terms are due to a remaining shift which we will calculate next.

To find the translation we take the derivative of Equation 4.17 w.r.t $\mathbf{v}_a$ giving:

$$2\mathbf{\Lambda}\mathbf{v}_a + \mathbf{b}_a \tag{4.18}$$

If the unknown translation is $\mathbf{t}$ the Equation 4.18 will vanish at the origin (i.e. at $\mathbf{v}_a = \mathbf{0}$ ) after this translation.

$$\begin{aligned} 2\mathbf{\Lambda}(\mathbf{v}_a + \mathbf{t}) + \mathbf{b}_a &= 0 & (4.19)\\ \mathbf{v}_a = \mathbf{0} \Rightarrow 2\mathbf{\Lambda}\mathbf{t} &= -\mathbf{b}_a & (4.20)\\ \Rightarrow \mathbf{t} &= -\frac{1}{2}\mathbf{\Lambda}^{-1}\mathbf{b}_a & (4.21) \end{aligned}$$

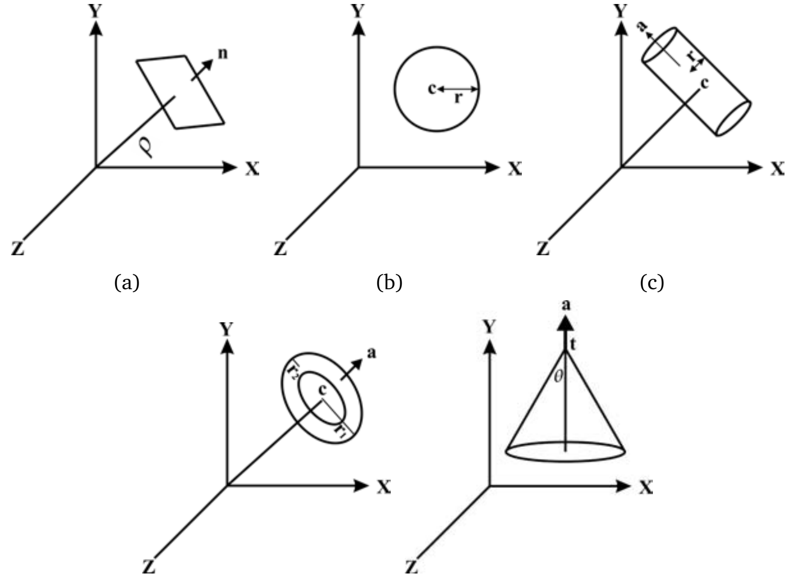
Now, by replacing $\mathbf{v}_a$ by $\mathbf{v}_b = (\mathbf{v}_a + \mathbf{t})$ in Equation 4.17 we can get rid of the linear terms. As a result of this substitution, the final constant term $d$ is changed to $d_b = d + \mathbf{t}^T\mathbf{b}_a$.

The final equation of the quadric in canonical form is given by:

$$\mathbf{v}_b^T\mathbf{\Lambda}\mathbf{v}_b + d_b = 0 \tag{4.22}$$

This gives an excellent method to get approximate values for the orthogonal distance based non linear least squares method. The rotation matrix equals $\mathbf{E}^T$ and the translation is given by $\mathbf{t}$.

The canonical analysis of the fitted quadric (Equation 4.12) leads to approximate values for rotation, translation and other shape parameters. We will use this technique of approximate value calculation for spheres, cylinders, and cones. For approximate value calculation of the torus the fitting of a quartic surface is very noise-sensitive and a piece-wise cylinder fitting based procedure will be used instead. The details of this method will be given in Section 4.3.5.



(a) (b) (c)

(d) (e)

**Figure 4.2:** *Parameters of the object models (a) The plane* $\mathbf{\Pi}$ *is parametrized by its normal vector* $\mathbf{n} = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}$ *and perpendicular distance from the origin* $\rho$ *(b) A sphere* $\mathbf{S}$ *is parametrized by its center* $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}$ *and its radius* $r$ *(c) The cylinder is represented by its axis* $\mathbf{a} = \begin{pmatrix} a_x & a_y & a_y \end{pmatrix}$*, the point on axis closest to origin* $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}$*, and the radius* $r$ *(d) A torus is given by* $\mathbf{a} = \begin{pmatrix} a_x & a_y & a_z \end{pmatrix}$ *for the axis direction,* $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}$ *for the central point, the major radius* $r_1$ *and the minor radius* $r_2$*. (e) A cone is given by its apex* $\mathbf{t}$*, axis direction* $\mathbf{a}$ *and the angle with the vertical* $\theta$*.*

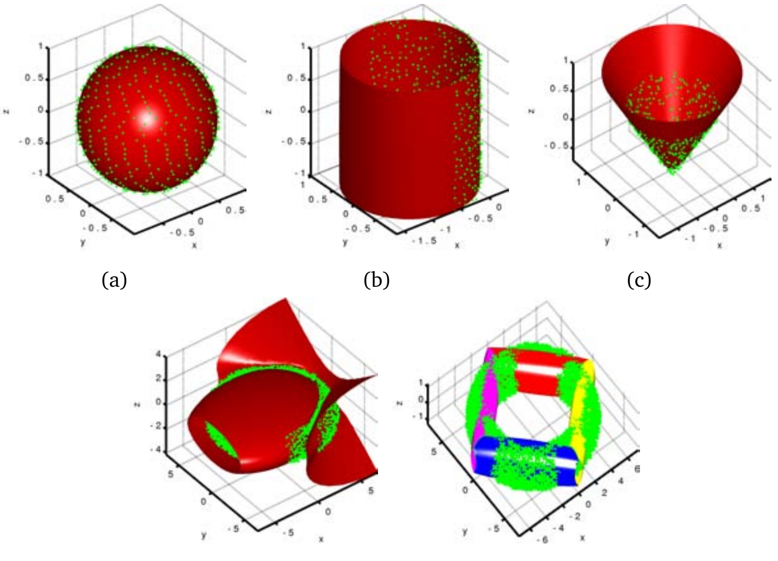
### 4.3.2 Plane fitting

We described a method for fitting planes based on the eigen analysis in Chapter 2. Here we present another method for fitting planes through non-linear least squares. This method can use the covariance matrix of the input data and propagate it to the variance of the final estimated parameters through Equation 4.7. As a result, along with the plane parameters their covariances are also estimated. The method from Chapter 2 can still be used for getting the approximate values.

A plane $\mathbf{\Pi}$ is parametrized by its normal vector $\mathbf{n} = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}$ and the perpendicular distance from the origin $\rho$ (Figure 4.2(a)). As there are only three degrees of freedom for a plane, there is a constraint on the length of the normal vector which must be one i.e. $\|\mathbf{n}\| = 1$. The distance of a given point $\mathbf{p} = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}$ from the plane is given by:

$$d_{\Pi} = \langle \mathbf{n}, \mathbf{p} \rangle - \rho \tag{4.23}$$

As stated earlier, the minimization of the sum of the squares of these distances i.e. $\sum d_{\Pi}^2$, requires the partial derivatives of $d$ with respect to all the shape



(a) (b) (c)

(d) (e)

**Figure 4.3:** *Approximate values through quadric fitting (a) sphere (b) cylinder (c) cone (d) torus approximation through quartic fitting is sensitive to noise (e) torus through piecewise quadric fitting.*

parameters of the object under consideration (Section 4.2). The expressions for these derivatives $\frac{\partial d}{\partial n_x}, \frac{\partial d}{\partial n_y}, \frac{\partial d}{\partial n_z}, \frac{\partial d}{\partial \rho}$ in a more general form (including a rigid transformation) are derived in a later chapter on registration (Chapter 5).

### 4.3.3 Sphere fitting

A sphere $\mathbf{S}$ is parametrized by its center $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}$ and its radius $r$ (Figure 4.2(b)). The distance of a point $\mathbf{p}$ from the sphere is given by

$$d_S = \|\mathbf{p} - \mathbf{c}\| - r = \sqrt{(\mathbf{p} - \mathbf{c})^T (\mathbf{p} - \mathbf{c})} - r \tag{4.24}$$

The partial derivatives required for minimizing this orthogonal distance are derived in Chapter 5.

For the calculation of the approximate values, we can use quadric fitting followed by canonical analysis (Section 4.3.1). Another approximation based on the implicit equation of the sphere is given by Bookstein (1979).

### 4.3.4 Cylinder fitting

Cylinders are one of the most commonly encountered geometric objects on industrial sites. We have presented an extension of the Hough transform for their automatic detection in Chapter 3. Once a cylinder has been detected and the points belonging to it have been selected through the Hough transform, the final estimation of the parameters is done through the least squares method of this section.

The cylinder is one of the quadric surfaces (Table 4.2). Its equation in standard position (centered at the origin, axis along the z-axis) is given by

$$x^2 + y^2 - r^2 = 0 \tag{4.25}$$

If cylinder fitting is being done as a separate step and not as a follow up to the Hough transform, we need good approximate values. One such method for approximating the parameters of the cylinder is through quadric fitting. The estimated quadric is then converted to the standard form of Equation 4.25 through the method of Section 4.3.1. An example is shown in Figure 4.3.

Another method for the approximate estimation is based on using the behavior of the normals belonging to the cylindrical objects. As shown in Chapter 3 the normals of a cylinder make a great circle on the Gaussian sphere. We can estimate the orientation by fitting a plane to the estimated normals of the given cylindrical point cloud. The normal direction of the fitted plane approximates the cylinder axial direction.

For fitting a cylinder to the points in the least squares sense, the orthogonal distance of the point from the cylinder surface is required. For this purpose we use an over-parametrized formulation as it leads to more compact expressions for the partial derivatives. The cylinder is represented by its axis $\mathbf{a} = \begin{pmatrix} a_x & a_y & a_y \end{pmatrix}$, the point on axis closest to the origin $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}$, and the radius $r$ (Figure 4.2(c)). As there are only five degrees of freedom for the cylinder, we have two constraints. Firstly, the length of the axial direction vector must be one i.e. $\|\mathbf{a}\| = 1$. Secondly, the point $\mathbf{c}$ must be closest to the origin, which means that $\mathbf{c}$ and $\mathbf{a}$ should be perpendicular ($\mathbf{a} \perp \mathbf{c} \Rightarrow \langle \mathbf{a}, \mathbf{c} \rangle = 0$). The geometric distance of a point from the surface of the cylinder is given by

$$d = \|(\mathbf{p} - \mathbf{c}) \times \mathbf{a})\| - r \tag{4.26}$$

The partial derivatives of $d$ w.r.t cylinder parameters are derived in Chapter 5

### 4.3.5 Torus fitting

A torus is usually not found in a complete form on industrial sites. Usually it appears as a bend or some other subpart of a more complex object.

In its standard form (centered on the origin with its axis along z-axis) a torus can be represented by the following quartic surface.

$$(x^2 + y^2 + z^2 - (r_1^2 + r^2))^2 - 4r_1^2(r_2^2 - z^2) = 0 \tag{4.27}$$

where $r_1$ is the major axis and $r_2$ the minor axis. When the torus has undergone a general rigid body transformation Equation 4.27 changes to a full quartic equation with 35 terms. A quartic can be fitted to a given set of points using linear least squares in a way similar to the one presented in Section 4.3.1 for the quadric fitting. However, with 34 degrees of freedom the solution of quartic fitting is extremely sensitive to noise. An example of this sensitivity is shown in Figure 4.3(d). Additionally, the approximation of the geometric distance by the algebraic distance for a quartic polynomial is much worse than for a quadric. Keeping these reasons in view we do not use quartic fitting for torus approximation.

For approximating the parameters of a torus we fit piecewise cylinders to the point cloud. Provided there are enough axial lines to determine a plane (whose normal approximates torus orientation) we can approximate the parameters of the torus. An example of this procedure is shown in Figure 4.3(e).

For least squares fitting we use an over-parametrized representation consisting of $\mathbf{a} = \begin{pmatrix} a_x & a_y & a_z \end{pmatrix}$ for the axis direction, $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}$ for the central point, $r_1$ for the major radius and $r_2$ the minor radius (Figure 4.2(d)). If the length of the axis direction is constrained to unity ($\|\mathbf{a}\| = 1$), the orthogonal distance $d$ of a point $p = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}$ from the surface of the torus is given by:

$$d = \sqrt{(\|(\mathbf{p} - \mathbf{c}) \times \mathbf{a}\| - r_1)^2 + \langle(\mathbf{p} - \mathbf{c}), \mathbf{a}\rangle} - r2 \tag{4.28}$$

The partial derivatives of $d$ w.r.t torus parameters are derived in Chapter 5

### 4.3.6 Cone fitting

A cone is given by the coordinates of its top $\mathbf{t} = \begin{pmatrix} t_x & t_y & t_z \end{pmatrix}$, direction of axis $\mathbf{a} = \begin{pmatrix} a_x & a_y & a_z \end{pmatrix}$ and the slope of the radius change, measured as an angle in radians from the central line given by $\theta$ (Figure 4.2(e)).

To find the distance of a given point $\mathbf{p}$ from the cone, we have to operate on the triangle formed by $\mathbf{p}$, $\mathbf{t}$ and the closest point $\mathbf{p}_c$. The location of the closest point $\mathbf{p}_c$ is not known, but according to the definition of the orthogonal distance $(\mathbf{p} - \mathbf{p}_c) \perp (\mathbf{t} - \mathbf{p}_c)$. Through simple trigonometry it can be proved that the distance of $\mathbf{p}$ from the cone is given by

$$d = \|\mathbf{h}\| \sin(\cos^{-1} \langle \frac{\mathbf{h}}{\|\mathbf{h}\|}, \mathbf{a}\rangle - \theta) \tag{4.29}$$

where $\mathbf{h} = \mathbf{p} - \mathbf{t}$.

For getting approximate values for cone fitting we use quadric fitting followed by canonical analysis(Section 4.3.1). An example of this procedure is shown in Figure 4.3(c).

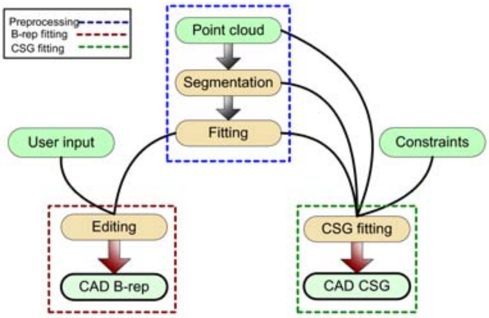


**Figure 4.4:** *Comparison of CSG and B-rep fitting. While the conversion from the surface patches to a B-rep requires extensive manual editing, there is no such editing stage for CSG fitting.*

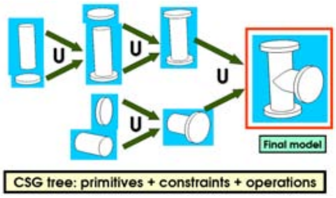


**Figure 4.5:** *Constructive solid geometry (CSG) represents industrial models through boolean operations on simple geometric primitives like boxes, cylinder, spheres and cones. Here a flanged T-junction is represented by the union of 5 cylinders*

## 4.4 CSG Fitting

In this section we present methods for fitting CAD models represented as constructive solid geometry trees to the point clouds. Most of the previously reported methods for CAD model fitting (Ahn et al., 2002; Fisher, 2002; Várady et al., 1997; Werghi et al., 1999b) have focused on recovering patches of simple geometric surfaces, which are then connected together to make a B-rep model. These patches are recovered using segmentation (Hoover et al., 1996; Jiang et al., 2000) followed by fitting of simple surface models (Petitjean, 2002). Alternatively B-Splines and NURBS based surface patches can be fitted (Ko et al., 2003). As all the patches necessary for the recovery of the complete model are seldom fully visible in the captured data, this fitting must be followed by an editing stage, where the remaining gaps at the edges and the joints are filled (Figure 4.4). This editing is mostly manual and its quality depends on the skill of the operator. Furthermore, as this final stage does not involve any parameters estimation, the final B-rep model does not contain any associated quality measures.

Constructive solid geometry (CSG) is a way to represent CAD models through boolean operations on simple primitives like spheres, cones, cylinders, tori and boxes (Mortenson, 1985; Requicha and Voelcker, 1982). Some examples of the boolean operations are union, intersection, and difference. As an example a flanged T-junction can be represented as a CSG object by using the union operation on five cylinders (Figure 4.5).

### 4.4.1 Motivation for using CSG

The approach for fitting CSG models for industrial reconstruction is motivated by the following observations:

**Compact representation** Although CSG is not as general purpose as B-rep, whenever applicable it is more compact in terms of the information required for designing the same model. Moreover, when fitting a model non compactness of B-rep leads to an explosion in the number of parameters, all of which need to be estimated. To take a simple example a box in B-rep needs 6 bounding planes, each with its enclosing bounding rectangle along with the constraints that the faces should be orthogonal and must intersect. Encoding of all this information in the form of a nonlinear least squares problem is quite involved. In contrast, a box is one of the standard primitives in CSG and can be represented by 9 parameters, 3 for the position, 3 for the rotation (quaternion with one constraint), and three for the sizes of the box along x, y, and z directions. Compared to B-rep this is more compact and its conversion to a nonlinear least squares problem is quite direct.

**Natural encoding of constraints** In B-rep the basic building block is a surface with its associated boundary, and all constraints must be specified between the surface patches. In contrast in CSG the whole primitive (e.g. a box or a cylinder) is the basic building block, and the specification of constraints is also in terms of these primitives which is more natural for industrial parts. For example the relationships among the sub-parts of a flanged T-junction in Figure 4.5 can be specified in terms of the collinearity and the orthogonality of the constituent cylinders. Similarly the relationships between the relative positions and the radii can be easily encoded as a constraints. Specifying the same geometric information in a B-rep structure would be more difficult.

**No manual editing** Due to the complexity of formulating B-rep fitting problem in a least squares framework, most approaches in the literature consist of segmentation, followed by fitting of simple primitives to the segmented point cloud and a final editing and refinement stage. This editing is usually done manually by a human operator. In contrast in our presented approach of CSG fitting no final editing is necessary. If some information is missing then the required results can be obtained by specifying more constraints that provide the missing information, and repeating the fitting procedure. A comparison of the work flow for CSG vs B-rep based modeling is shown in Figure 4.4. There we see that the B-rep based approaches involve a manual editing stage while CSG based methods estimate the complete model through fitting using the measured data and a priori information coded as constraints.

**Quality measures for fitting** As CSG fitting is based on a nonlinear least squares framework, without any manual editing involved, the final result contains the covariance matrix of the estimated parameters (Equation 4.7). In contrast, because of the manual editing stage, B-rep fitting does not provide any quality measures of the estimated parameters of the final models.

**Use of engineering libraries** In many industries the CAD libraries used during site design are based on CSG. For example in the petrochemical industries most of the available CAD design packages use CSG as an internal representation format (AVEVA PDMS, 2005; Intergraph PDS, 2005). These libraries also contain information about the geometric and parameter constraints on the constituent sub-parts of the CAD model. By using CSG for fitting we ensure that the resultant model can be easily converted to one of the industry standard formats and thus provide a directly usable as-built CAD model.

**Extensible to images** As noted in Chapter 1 images provide a complementary source of information to the point clouds. The use of CAD models for fitting to images was pioneered by Lowe (1991). He estimated the pose and shape parameters by minimizing the distance of the visible edges from the hidden-line projection of the estimated model. An extension of this approach for fitting CSG objects to image gradients and point measurements for industrial reconstruction has been reported by Ermes et al. (1999); Tangelder et al. (1999, 2003), where geometric constraints are also used. An advantage of using CSG fitting to the point clouds is its extensibility to include images when they are available. In Chapter 7 we extend the CSG fitting techniques presented here to include image measurements and show that it leads to better estimation quality. Additionally, we provide a taxonomy of constraints along with the details of their enforcement through constrained least squares. In the traditional way of B-rep fitting the final stage of the conversion from surface patches to the B-rep model does not involve any estimation and thus images cannot be easily included there.

### 4.4.2 Distance Approximation Algorithms

The key problem in fitting a given point cloud to a selected CSG model is to calculate the orthogonal distance of a point from the model surface ($\Psi$ in Equation 4.2). Moreover, the partial derivatives of the distance with respect to the shape and the pose parameters of the CSG model are needed for the non-linear least squares.

We present three different methods to solve this problem. We name these methods *ICS* (Iterative Closest Surface-point), *ICT* (Iterative Closest Triangle-point), and $ICP^2$ (Iterative Closest Point-cloud-point). These methods convert a given CSG model to a B-Rep, a triangle mesh, and a point cloud respectively to calculate the orthogonal distance of a given point from the model surface. These methods provide different trade-offs between speed and accuracy and are compared in Section 4.5.

For many unbounded objects like infinite planes, infinite cylinders, spheres etc. it is possible to get closed-form expressions for the orthogonal distance as well as its partial derivatives. As the CSG uses bounded objects and also employs boolean operations among them, it is very difficult to derive an analytical expression for the distance of a point from an arbitrary CSG object. As a result we have to use

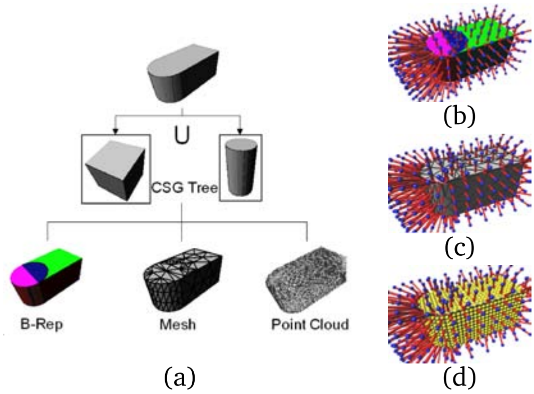


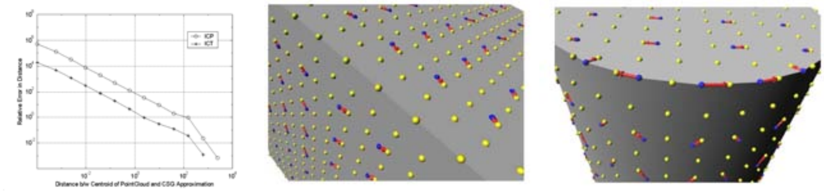
**Figure 4.6:** *Three approximation methods for calculating orthogonal distance from a CSG tree (a) A CSG object and three different approximation methods (b) ICS - Iterative closest surface point (c) ICT - Iterative closest triangle point (d) ICP$^2$ - Iterative closest point-cloud point*

numerical methods to calculate the distance as well as its partial derivatives. We present the following three methods to solve this problem (Figure 4.6(a)):

**Iterative Closest Surface-point (ICS)** In this method the CSG model is converted to a B-rep and the distance of a given point to the closest surface in the B-rep is calculated (Figure 4.6(b)). We call this method Iterative Closest Surface-point (**ICS**). For this method we use ACIS (A commercial geometric modeling package (Spatial, 2004)) for converting a CSG object to its B-rep as well as for calculating the distance of the points from the closest boundary surface. This method is geometrically most accurate as it uses the exact mathematical formulation of the surface and its boundaries which is limited only by the numerical precision of the computer. However, this comes at a high computational cost making it the slowest method.

**Iterative Closest Triangle-point (ICT)** This method approximates the CSG model with a triangulated mesh and calculates the distance of a given point from the closest triangle (Figure 4.6(c)). We name this method Iterative Closest Triangle-point (**ICT**). For the conversion of the CSG to the triangular mesh ACIS is used. After this conversion, the distance of the given point from the CSG object is approximated by its distance to the closest triangle in this mesh. In 2D ICT can be compared to approximating a general curve by a set of linear segments. When the points are very close to the surface the relative difference between the distances approximated by ICT and ICS can be significant but this difference rapidly falls as we move away from the surface. The difference also depends on the type of the surface and the number of triangles used in the mesh. For the objects made of planar faces (e.g. a box) the results of ICT are as accurate as those of ICS. But for curved surfaces like spheres and cylinders the performance of ICT depends the number of triangles used to approximate the object.

**Iterative Closest Pointcloud-point ( ICP$^2$ )** This method approximates a given CSG model by a point cloud and returns the distance of a given point to the closest point contained therein (Figure 4.6(d)). We call this method Iterative Closest Point-cloud-point ( **ICP$^2$** ). For ICP$^2$ the CSG object is converted to a



(a) (b) (c)

**Figure 4.7:** *Errors in ICP$^2$ and ICT (a) The relative error in ICT and ICP$^2$ as a function of the distance from the center of the object (b) ICP$^2$ has errors even for planar faces (c) For curved surfaces ICP$^2$ becomes even worse.*

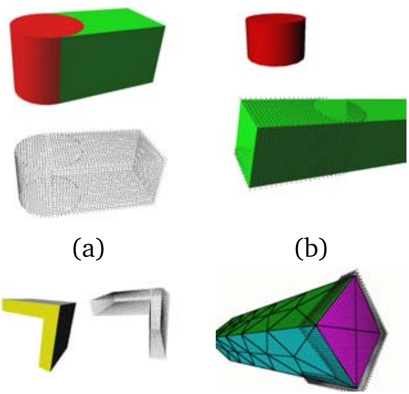
> point cloud, and the distance of the closest point therein to a given data point is used to approximate the distance to the CSG object. The accuracy of this approximation depends on the point density used as well as on the type of the surface. Compared to ICT, which is a first order approximation, ICP$^2$ is a zeroth order approximation. As a result, even for planar surfaces the distance has residual errors. In terms of speed this method is found to be the fastest as a kd-tree is used for searching the nearest neighbor (Arya et al., 1998). The ICP$^2$ we present here has some similarities to Iterative Closest Point (Besl and McKay, 1992; Rusinkiewicz and Levoy, 2001) which is widely used for the registration of point clouds. However, the problem of fitting is more complex than that of registration, as it involves the simultaneous estimation of the rigid body transformation as well as the shape parameters of the subparts of the CSG tree.

These methods decrease in computational complexity as well as accuracy from top to bottom.

To compare the numerical accuracy of ICT and ICP$^2$ against ICS, we define the average relative error as follows:

$$\mathrm{E}_r^{ICX} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\Psi_i^{ICX} - \Psi_i^{ICS}}{\Psi_i^{ICS}}\right| \tag{4.30}$$

A plot of the average relative error for ICT and ICP$^2$ with respect to the distances between the centroids of the point cloud and that of the CSG object is shown in Figure 4.7(a). The plot uses cylinder-box-union (Figure 4.6) as a test object. We see that the relative error for both methods is inversely proportional to the distance between the centroids of the measured point cloud and the current estimated CSG object. This difference becomes significant only when points are very close to the CSG surface. This means that ICS is necessary only for the final iterations. For the initial iterations, when the estimated solution is far from the minimum, ICT or ICP$^2$ can be used. Furthermore, as expected, for a given point density the average relative error of ICP$^2$ is higher than that of ICT.



(a) (b)

(c) (d)

**Figure 4.8:** *Importance of constraints for CSG fitting (a) Before fitting (b) without constraints the geometric relationships of the box and the cylinder are not ensured (c) L-junction before fitting (d) without constraints the relationship of boxes is easily violated*

### 4.4.3 Internal Constraints

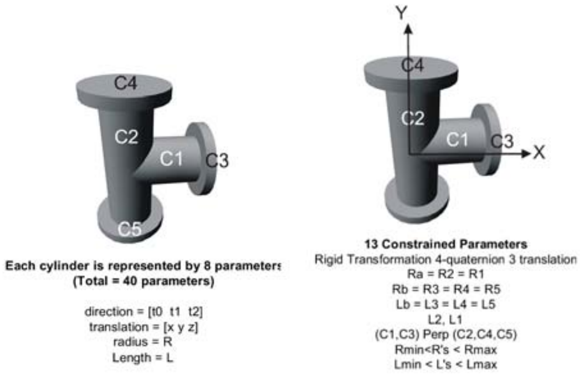
One of the attractive features of CSG is its ability to compactly represent a large number of complex objects by applying boolean operations to a few simple geometric primitives. This powerful feature has its downside for an iterative optimization process, as in the absence of extra geometric information it is very easy to get trapped in the local minima. In most of the CAD models this extra geometric information is implicit and is not encoded as a part of the design. But for fitting this geometric information must be encoded explicitly as it can reduce the degrees of freedom, and thus can avoid the local minima arising from unwanted configurations.

In Figure 4.8 we show two fitting scenarios without employing the internal constraints. As is clear from Figure 4.8(a) and 4.8(c) initial values are not very good. Consequently, in the absence of any geometric constraints, the least squares fitting procedure assigns all the points to one of the primitives, and the other primitive remains almost fixed in the initial position. Thus, we find a solution coming from one of the many local minima. However, this is not the desired solution, as we would like to enforce the geometric and topological relations at all stages of the fitting process. We can avoid this situation by enforcing *Internal constraints* (Tangelder et al., 2003), which encode the geometric relationships between the constituent primitives of a CSG tree. For example in the case of cylinder-box-union (Figure 4.8(a)) we would like the diameter of the cylinder to equal one of the sizes of the box. Furthermore, the axis of the cylinder and the box must be parallel and the cylinder must always intersect the box.

Similarly for a flanged T-junction consisting of the union of five cylinders, by enforcing all the geometric relationships explicitly the number of free parameters is reduced from 40 to 13 [1] (Figure 4.9). The reduction in the number of the free parameters for different CSG models due to the enforcement of constraints is

---

[1] This includes the over-parametrization for the quaternion as the constraint $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ cannot be enforced.



(a) (b)

**Figure 4.9:** *The reduction in the parameters of the CSG object by using constraints (a) without constraints the flanged T-junction needs 40 parameters (b) with constraints the parameters are reduced to 13. Additionally, the use of constraints ensures that the final result conforms to the design intent.*

shown in Table 4.3.

The same argument can be applied to the valid ranges for the values of different shape parameters, also called *bound constraints*. When there is some a priori information available about the lower and upper limits it should be used during the fitting stage. Furthermore, these bound constraints can be utilized to avoid geometrically meaningless values. For example, the radius of a cylinder and the sizes of a box can be enforced to be always positive.

The standard solution for implementing the constraints is to use Constrained optimization. In this chapter we use explicit constraint enforcement, and solve the resulting unconstrained optimization problem. The concept of explicit *Internal constraints* can be explained by an example. A box is represented by ten parameters in our primitive library, four parameter are for the quaternion representation of the rotation, three for the translation, and three for the x, y, and z sizes. If we do not enforce any constraints for the box-minus-box case we have twenty parameters giving us many erroneous solutions like the one shown in Figure 4.8(d). By looking closely at the design intent, the second box can be represented just by the width of the resulting L-shaped hollow frame. The rest of the parameters can be deduced from the first box. This reduces the number of parameters from 20 in the unconstrained case to 11 in the constrained case. In addition we can enforce the maximum and the minimum bounds on the sizes of the boxes as well as on the values of the other parameters if we have a priori information about them. Table 4.3 gives a comparison of the number of parameters for different CSG objects for both constrained and unconstrained cases.

Although, the concept of explicit constraint enforcement enables us to use unconstrained least squares for solving constrained problems, it has some serious limitations. It can only deal with linear constraints which have to be explicitly solved for each new object. This requires a new function that takes the minimal number of parameters and converts them to an expanded set of parameters needed to instantiate a CSG object. Furthermore, the external constraints which relate one CSG tree to another (for example the connection of one of the cylinders from one

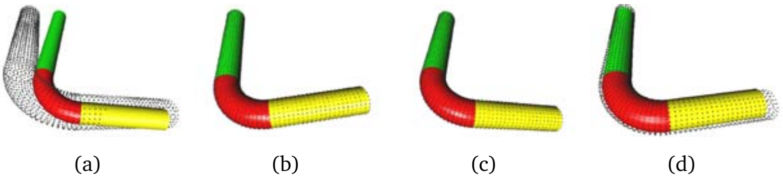

(a) (b) (c) (d)

**Figure 4.10:** *Results of fitting an L-junction (a) before fitting (b) after ICS (c) after ICT (d) after ICP$^2$ . Note the remaining error*

CSG object to the cylinder from another CSG object) cannot be specified. Similarly, only those constraints which can be explicitly solved can be incorporated. This means that nonlinear constraints like the one on the quaternion length cannot be included in the estimation process. These problems are addressed in Chapter 7 where we use constrained optimization techniques to fit CSG models to both point clouds and images.

### 4.4.4 Fitting Algorithm

The algorithm for CSG fitting is quite similar to the one given in Section 4.2.1. To estimate the partial derivatives $\frac{\partial \Psi_{ICX}}{\partial \tau_i}$ we used finite differences. As noted by Dennis and Schnabel (1996) for a sufficiently small step-size the results obtained from the finite difference approximation of the partial derivatives for the least squares problem are indistinguishable from the ones obtained by using the analytical derivatives. Of course the errors inherent in the distance calculation for ICP$^2$ and ICT are also propagated to the partial derivative estimation.

We use quaternions to specify the rotation parameters (Shoemake, 1985), which have an extra constraint that the sum of the squares of their four elements should equal one. This cannot be enforced due to our use of unconstrained optimization. This often results in an ill-conditioned or sometimes a singular system. To tackle such conditions we solve Equation 4.3 using singular value decomposition (Golub and Loan, 1991) which identifies the linearly dependent columns and removes them from the system.

To enforce the bound constraints explicit enforcement is used. At the end of each iteration all bound constraints are checked, and in case of a violation the value of the violating parameter is clipped to the specified bound.

## 4.5 Results

We tested the three methods outlined above on a few CSG objects, and compared the results on the basis of rms error of fit, iterations required, and time taken. The results are presented in Table 4.3.

The fitting tests were conducted for three different data sets for each object; the first one was with no noise, the second one contained zero mean Gaussian noise with standard deviation of 2mm added to each of x, y and z coordinates of the point cloud. The noise level in the third set had a standard deviation of 5mm. For each dataset we show the rms error between the final estimated model and the input point cloud. This error is calculated using ICS.

Comparing on the basis of speed, we find that on average ICS is 20 to 200 times slower than ICT. ICP$^2$ is usually faster than ICT, but the final error as well as the number of iterations required is higher. This can be explained by the erroneous estimation of the partial derivatives. Even though the number of points used for ICP$^2$ is greater than the number of triangles for ICT, small perturbations required by the finite differences combined with the non-smooth nature of ICP$^2$ results in noisy partial derivatives. Due to the same reason sometimes ICP$^2$ gets trapped in a local minimum resulting in a higher rms error compared to ICS and ICT (Figure 4.10(d)).

As noted earlier, the error in the distance approximation by ICT depends on the type of the surface as well as on the number of the triangles used. For objects (a) and (f) in Table 4.3, the results obtained by ICT and ICS have the same amount of error because they are composed of planar faces. In contrast, for objects (d) and (e) the error of ICT is significantly higher compared to ICS, because the objects are composed of curved surfaces.

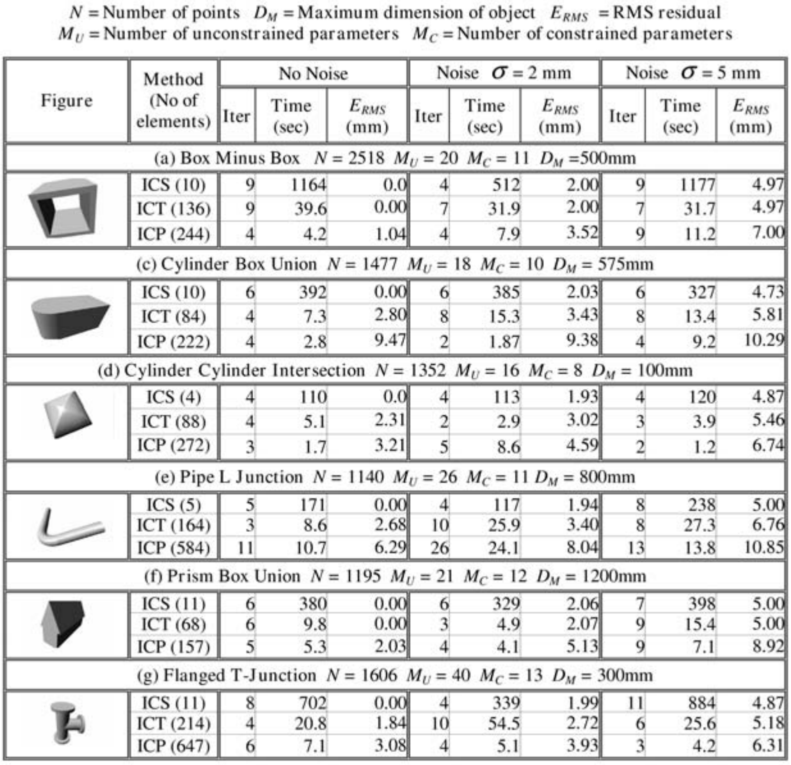
The higher speed of ICT compared to ICS and the better and smooth performance compared to ICP$^2$ makes it the method of choice, giving the best trade-off between the speed and the accuracy. Additionally, as seen from the plot of the average relative error in Figure 4.7(a), for the initial iterations ICT can be safely used, as its average relative error is very small. For the final iterations either ICS or ICT with a higher number of triangles should be employed.

The Table 4.3 also shows that the explicit enforcement of the internal geometric and bound constraints leads to a significant reduction in the number of parameters. Moreover, erroneous configurations are also successfully avoided, because at the end of each iteration only valid solutions are accepted.

The speed comparison of ICT with ICP$^2$ does not reflect the true situation; as for ICP$^2$ we use kd-trees while for ICT an exhaustive search is used. By employing a faster search methods along with multi-resolution techniques we expect the speed performance of ICT to come much closer to that of ICP$^2$ .

## 4.6 Conclusions

In this chapter we have presented methods for fitting both simple and CSG models to point clouds. The fitting was posed as an optimization problem and solved through a nonlinear least squares method. Methods to get good approximate values were presented based on quadric fitting and its canonical analysis. We



$N$ = Number of points $D_M$ = Maximum dimension of object $E_{RMS}$ = RMS residual
$M_U$ = Number of unconstrained parameters $M_C$ = Number of constrained parameters

| Figure | Method (No of elements) | No Noise | | | Noise $\sigma = 2$ mm | | | Noise $\sigma = 5$ mm | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Iter | Time (sec) | $E_{RMS}$ (mm) | Iter | Time (sec) | $E_{RMS}$ (mm) | Iter | Time (sec) | $E_{RMS}$ (mm) |
| (a) Box Minus Box $N = 2518$ $M_U = 20$ $M_C = 11$ $D_M = 500$mm | | | | | | | | | | |
| | ICS (10) | 9 | 1164 | 0.0 | 4 | 512 | 2.00 | 9 | 1177 | 4.97 |
| | ICT (136) | 9 | 39.6 | 0.00 | 7 | 31.9 | 2.00 | 7 | 31.7 | 4.97 |
| | ICP (244) | 4 | 4.2 | 1.04 | 4 | 7.9 | 3.52 | 9 | 11.2 | 7.00 |
| (c) Cylinder Box Union $N = 1477$ $M_U = 18$ $M_C = 10$ $D_M = 575$mm | | | | | | | | | | |
| | ICS (10) | 6 | 392 | 0.00 | 6 | 385 | 2.03 | 6 | 327 | 4.73 |
| | ICT (84) | 4 | 7.3 | 2.80 | 8 | 15.3 | 3.43 | 8 | 13.4 | 5.81 |
| | ICP (222) | 4 | 2.8 | 9.47 | 2 | 1.87 | 9.38 | 4 | 9.2 | 10.29 |
| (d) Cylinder Cylinder Intersection $N = 1352$ $M_U = 16$ $M_C = 8$ $D_M = 100$mm | | | | | | | | | | |
| | ICS (4) | 4 | 110 | 0.0 | 4 | 113 | 1.93 | 4 | 120 | 4.87 |
| | ICT (88) | 4 | 5.1 | 2.31 | 2 | 2.9 | 3.02 | 3 | 3.9 | 5.46 |
| | ICP (272) | 3 | 1.7 | 3.21 | 5 | 8.6 | 4.59 | 2 | 1.2 | 6.74 |
| (e) Pipe L Junction $N = 1140$ $M_U = 26$ $M_C = 11$ $D_M = 800$mm | | | | | | | | | | |
| | ICS (5) | 5 | 171 | 0.00 | 4 | 117 | 1.94 | 8 | 238 | 5.00 |
| | ICT (164) | 3 | 8.6 | 2.68 | 10 | 25.9 | 3.40 | 8 | 27.3 | 6.76 |
| | ICP (584) | 11 | 10.7 | 6.29 | 26 | 24.1 | 8.04 | 13 | 13.8 | 10.85 |
| (f) Prism Box Union $N = 1195$ $M_U = 21$ $M_C = 12$ $D_M = 1200$mm | | | | | | | | | | |
| | ICS (11) | 6 | 380 | 0.00 | 6 | 329 | 2.06 | 7 | 398 | 5.00 |
| | ICT (68) | 6 | 9.8 | 0.00 | 3 | 4.9 | 2.07 | 9 | 15.4 | 5.00 |
| | ICP (157) | 5 | 5.3 | 2.03 | 4 | 4.1 | 5.13 | 9 | 7.1 | 8.92 |
| (g) Flanged T-Junction $N = 1606$ $M_U = 40$ $M_C = 13$ $D_M = 300$mm | | | | | | | | | | |
| | ICS (11) | 8 | 702 | 0.00 | 4 | 339 | 1.99 | 11 | 884 | 4.87 |
| | ICT (214) | 4 | 20.8 | 1.84 | 10 | 54.5 | 2.72 | 6 | 25.6 | 5.18 |
| | ICP (647) | 6 | 7.1 | 3.08 | 4 | 5.1 | 3.93 | 3 | 4.2 | 6.31 |

**Table 4.3:** *Comparison of ICS, ICT and ICP*$^2$

showed that CSG is a powerful and compact way for representing and fitting of complex models. The importance of constraints for getting reliable results by decreasing the degrees of freedom was shown through examples. Three different approximation methods for calculating the distance of a point from a CSG model and its partial derivatives with respect to the shape parameters were presented and compared based on speed and accuracy.

In Chapter 7 we will extend the CSG fitting to include images, and generalize the constraint enforcement by using constrained optimization.

Chapter 5

# Model-based registration

> Don't reinvent the wheel, just realign it.
>
> Anthony J. D'Angelo

Point cloud acquisition by using laser scanners provides an efficient way for 3D as-built modeling of industrial installations. Covering such an installation with point cloud data often requires data acquisition from multiple standpoints. Before the actual modeling can start the transformation parameters of all of these scans need to be determined. In this chapter we present two methods to register point clouds of industrial scenes with different coordinate definitions. Rather than using corresponding points whose determination is very error-prone, corresponding object models in different scans are used to determine the translation and rotation parameters of the scans. The first method, named *Indirect Method*, is a two-step approach as there object fitting and registration of the scenes is done separately. The second method, called *Direct Method*, simultaneously determines the shape and pose parameters of the objects as well as the registration parameters of all the scans. Both methods are designed such that optimal use can be made of the information available from the shapes present in industrial environments. Compared to Iterative Closest Point (ICP) the presented approach combines registration and modeling in one step and thus leads to better overall estimation of registration parameters. Furthermore, the simultaneous registration of multiple scans is possible which takes care of the problems arising from pair-wise scan registration. As the presented approaches are based on non-linear least squares, they provide quality measures in the form of a covariance matrix of the estimated parameters, which can be used to make decisions about the quality of registration. Results and comparison with ICP are presented on some point cloud data-sets from actual industrial sites.

## 5.1 Introduction

To model big industrial sites the first problem that occurs is defining one coordinate system for all point clouds. This means that for each point cloud in the project the relative orientation and position have to be found and possibly connected to the factory coordinate system. When a scene can be represented as two point sets with known correspondences, the rigid transformation consisting of six parameters (three rotation angles and three translation parameters) in a least square sense can be calculated using a closed form solution(Horn et al., 1988). As finding accurate corresponding points in point clouds is highly error-prone, that straightforward approach is usually not applicable.

The iterative closest point (ICP) algorithm (Besl and McKay, 1992) works without any pre-knowledge about the point-to-point correspondences. It establishes point-to-point correspondence iteratively based on the minimum Euclidean distance. If the initial values are good enough, this procedure usually converges. ICP is a pair-wise registration procedure, and cannot reliably handle simultaneous registration of multiple scans. This results in the propagation of registration errors as more scans are acquired and added to a project. Secondly a large overlap area is required for proper functioning of these algorithms. Typically an overlap area of 25–30 % is recommended for acceptable results. Furthermore, dealing with occlusions can be problematic, as ICP uses points without any consideration for the underlying geometry.

Various attempts have been made to make ICP more robust by filtering the correspondences for effective handling of occlusions (Guehring, 2001; Zhang, 1994). The wrong correspondences are detected by comparing with a dynamic distance threshold, leading to a better behavior in the presence of noise and occlusions.

ICP in its original form provides no measure about the precision and reliability of the estimated parameters. Some attempts have been made to remove this limitation. For example in (Guehring, 2001) statistical properties of the registration parameters are estimated during registration through ICP.

To improve the convergence properties of ICP many variations on Euclidean distance function have been proposed in the literature. Most of these algorithms do not establish point-to-point relationships, but instead look for the relationship between points in one set to the local surface-approximations of the points of the other data set (Chen and Medioni, 1992; Dorai et al., 1994). A variation of ICP for the simultaneous registration of more than two scans is presented in (Eggert et al., 1998b). To speed up closest point-on-surface search they use k-d trees. A comparison of different ICP variations is given in (Rusinkiewicz and Levoy, 2001). There they found point to the tangent plane distance to perform better than other distance variations of distance calculation.

A method for automatic registration of point clouds and images for the reconstruction of buildings is presented in (Stamos and Leordeanu, 2003). There

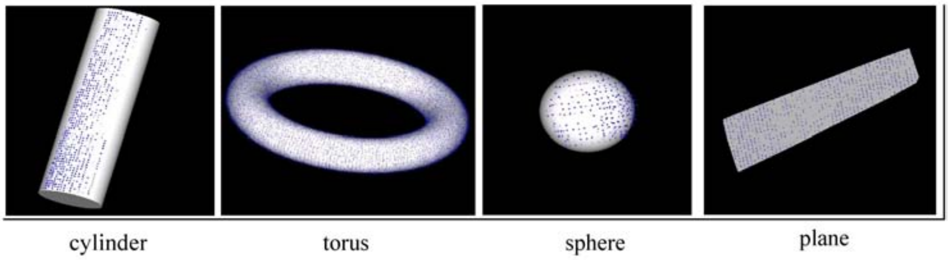


**Figure 5.1:** *Objects used as targets for registration*

planar faces are detected through segmentation of the point cloud. The intersection of these planes gives lines in 3D, which are used in a RANSAC-based (Fischler and Bolles, 1987) framework for range registration. Similarly, vanishing point detection in images and its relation with parallel lines in the point clouds is used for range-to-image registration. This framework exploits the domain specific knowledge to infer appropriate constraints, which results in high degree of automation for modeling of architectural scenes.

Many different approaches have been presented for the segmentation of a point cloud into planar and curved surfaces. A survey of such methods was presented in Chapter 1. Similarly Hough transform based techniques for automatic detection of planes and cylinders in point clouds were presented in Chapter 3. Using these methods detection and fitting of simple object models to point clouds can be automated with a great degree of success.

### 5.1.1 Presented approach

The approach we present here is based on the assumption that dense and accurate 3D point clouds of the industrial sites contain enough geometric information for the automatic detection and fitting of simple objects like planes, spheres, cylinders and tori. Rather than registering first and then modeling, we model first in each scan and then use the corresponding models to register the scans (Figure 5.1). We present two methods for this purpose:

**Indirect Method** This method minimizes the sum of squares of differences in the parameters of the corresponding models. This approach is geometrically approximate but much faster than the Direct method. This method has been previously described for planes and cylinders in (Dijkman and Heuvel, 2000). The second stage of the *Direct method* requires good approximate values, which can be obtained by using this method (Figure 5.2(a)).

**Direct Method** The Direct method leads to the simultaneous estimation of registration parameters of all scans and shape parameters of all objects used as targets. This is a non-linear least squares problem and the merit function

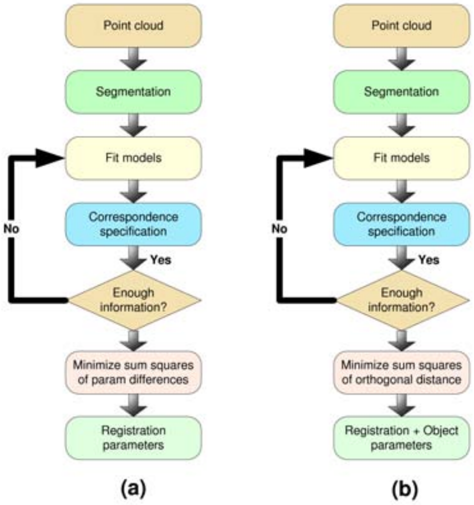

Point cloud
Segmentation
Fit models
Correspondence specification
No
Yes
Enough information?
Minimize sum squares of param differences
Registration parameters
(a)
Point cloud
Segmentation
Fit models
Correspondence specification
No
Yes
Enough information?
Minimize sum squares of orthogonal distance
Registration + Object parameters
(b)

**Figure 5.2:** *Two methods for registration (a) Indirect Method. For approximate registration through minimization of sum of squares of differences in parameters of corresponding models in different scans (b) Direct Method. For simultaneous registration of multiple scans by minimizing the sum of square distances of the points from the corresponding model surface.*

used is the orthogonal distance of the points coming from different scans from the surface of the object they belong to. This method consists of two sub-processes. The first one estimates only the registration parameters while the second sub-process named *Integrated Adjustment* results in simultaneous estimation of both registration and shape parameters. (Figure 5.2(b)). This can be seen as an extension of the concept of traditional *Bundle Adjustment* from the domain of images to that of point clouds.

### 5.1.2 Motivation

Our approach is motivated by the following observations:

**Consistent framework** The standard procedure of registration followed by modeling propagates the errors of the registration to the modeling step. Additionally there is no way to improve once the errors are more than some acceptable value as modeling and registration are based on two different paradigms. Our approach combines the model-fitting with the registration process, and uses one paradigm for both steps.

**Global registration** Standard registration approaches can only handle a pair of scans at a time. This means that information from only two overlaps can be used simultaneously. This leads to accumulation of errors and sub-optimal use of the available information. In contrast using our strategy *global registration* is possible, leading to better accuracy and optimal use of available data.

**Estimation of quality measures** As noted in (Bennamoun and Mamic, 2002) ICP just produces registration without giving any information about reliability and confidence of the estimated registration parameters. In contrast our approach is a direct application of least-square fitting and gives a full covariance matrix of the fitted parameters, which can be used to calculate the confidence region for all or a sub-set of the scan/object parameters.

**Less stringent overlap requirements** ICP requires overlap of up to 25–30 % to function properly, while the presented approach works even in cases where there is very small or no overlap between scans. The only requirement is that some part of corresponding object is visible in both scans. The reason is that compared to ICP which measures overlap in terms of overlapping points we measure overlap in terms of overlapping objects, and the resulting requirements are much more lenient especially for industrial sites. For example, consider the worst case scenario of a cylinder scanned from two opposite sides, ICP would not be able to proceed as there is no overlap in terms of points, while our algorithm can use the information from the geometry of the corresponding object for registration.

**Potential for automation** The presented approach can be automated by combining it with automatic correspondence search which can be based either on exhaustive search or on RANSAC (Fischler and Bolles, 1987). Automation of the registration process from ICP using RANSAC or similar techniques is bound to fail due to huge number of points involved. In the next chapter we present some strategies for doing such an automatic correspondence search through constraint propagation.

**Extensible to scan-to-image registration** ICP is a scan-to-scan registration method. As it uses point to point distance as a merit function it cannot work for registering images to scans. Although due to rapid progress in laser scanning the use of images for 3D measurement is going out of fashion, the fact remains that images are a complementary source of information. Images provide a better source of information on objects edges where point-clouds are highly unreliable and noisy. As shown in (Rabbani and Heuvel, 2004a) a combination of images and point clouds can result in better estimation accuracy. In contrast to ICP the presented object based registration method can be easily extended to image-to-scan registration through fitting of back-projected contours of the corresponding models to image edges(Ermes et al., 1999).

**Approximate values for complex models** The presented techniques can be used not only for the inter-scan registration, but also for the registration of a given model with an acquired point cloud during fitting. This provides a quick way to get approximate values for the fitting of complex models that can be carried out for example by the methods presented in (Rabbani and Heuvel, 2004b)

The rest of the chapter is structured as follows. In Section 5.2 we overview the used notation and give some commonly used mathematical expressions. Section 5.3 provides the details of the indirect method. The direct method is presented in Section 5.4. The last section presents the main results and conclusions.

## 5.2 Notation

Matrices are represented by bold capital letters e.g. $\mathbf{M}$. Vectors and points in $\mathbb{R}^3$ are represented by bold small letters e.g. $\mathbf{v}$. Small letters in normal font are used for scalars e.g. $s$. Normal font capital letters are used for objects e.g $S_1$ and $P_1$. For scans we will use $\Theta$ with appropriate subscripts.

For representing rotation parameters of the registration we use quaternion representation (Horn et al., 1988). Given a quaternion $\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$, the rotation matrix $\mathbf{R}$ can be formed as follows:

$$\mathbf{R} = \begin{bmatrix} 2\,{q_0}^2 + 2\,{q_1}^2 - 1 & 2\,q_1\,q_2 - 2\,q_0\,q_3 & 2\,q_1\,q_3 + 2\,q_0\,q_2 \\ 2\,q_1\,q_2 + 2\,q_0\,q_3 & 2\,{q_0}^2 + 2\,{q_2}^2 - 1 & 2\,q_2\,q_3 - 2\,q_0\,q_1 \\ 2\,q_1\,q_3 - 2\,q_0\,q_2 & 2\,q_2\,q_3 + 2\,q_0\,q_1 & 2\,{q_0}^2 + 2\,{q_3}^2 - 1 \end{bmatrix} \tag{5.1}$$

As rotation in $\mathbb{R}^3$ has only 3 degrees of freedom, the elements of quaternion vector $\mathbf{q}$ must be constrained by the following condition.

$$\|\mathbf{q}\| = \sqrt{{q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2} = 1 \tag{5.2}$$

The translation is represented by a vector $\mathbf{t}$:

$$\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T \tag{5.3}$$

In the following derivations we will often use $\frac{\partial \mathbf{R}}{\partial q_i}$ and $\frac{\partial \mathbf{t}}{\partial t_i}$, which can be evaluated by element-wise differentiation of Equation 5.1 and Equation 5.3 respectively.

## 5.3 Indirect Method

Because the industrial environments consist mainly of well-defined objects, the transformation parameters of different scans can be determined by aligning objects.

In this chapter we presents two methods to find the transformation parameters between different scans, using corresponding objects. The first method, *Indirect Method*, works toward the solution in the following three steps: (Figure 5.2):

1. Points are labeled as belonging to a certain object. This is done for several objects in different scans. This step can be automated by using a segmentation and object recognition algorithms (Chapter 2 & 3).
2. A least-squares based fitting algorithm calculates the object parameters for every object in each scan (Chapter 4).
3. The final transformation parameters are calculated by a least squares algorithm that has the object parameters as input. This step minimizes the

sum of square of the differences in object parameters as a function of scan transformation parameters.

This is a quick and computationally cheap method to get good approximate values. Mathematically it can be expressed as follows:

$$\min_{\{\mathbf{R}\},\{\mathbf{t}\}} \sum_{i=1}^{c} \sum_{j=1}^{m^i} \Delta_{u_{i_j}}^2 \tag{5.4}$$

Where

$$\Delta_{u_{i_j}} = \boldsymbol{\Psi}_{\mathbf{R}_i^a \mathbf{t}_i^a}(u_{i_j}^a) - \boldsymbol{\Psi}_{\mathbf{R}_i^b \mathbf{t}_i^b}(u_{i_j}^b)$$

In the Equation 5.4, we have $c$ correspondences, each of which consists of two objects $a$ and $b$. The model for $i-$th correspondence has $m^i$ parameters. $\boldsymbol{\Psi}_{\mathbf{R}_i^a \mathbf{t}_i^a}$ is an operator that applies the transformation of the scan, the object $a$ was modeled from, to its parameters $u_{i_j}^a$. We want to minimize the sum of squares of the differences in parameters of the corresponding object-pair $\Delta_{u_{i_j}}^2$. For this purpose we can adjust the values of transformation parameters of the scans $\{\mathbf{R}\}, \{\mathbf{t}\}$. Note the set notation, which means we might be dealing simultaneously with $N$ number of scans and hence their transformation parameters. The Equation 5.4 gives the expression for pair-wise registration. Although its extension to registration of multiple scans is straightforward we do not develop it in this paper, because the pair-wise formulation suffices for getting the approximate values.

The correlations between parameters of the model can have negative effects on the convergence of this procedure. This can be taken care of by weighting the equations by the inverse of the covariance matrix of model fitting.

To solve the non-linear least-squares problem in Equation 5.4 we will use Levenberg-Marquardt method (Björck, 1996; Press et al., 1988) as it gives the best convergence properties. For this purpose we will need partial derivatives of $\Delta_{u_{i_j}}$ w.r.t parameters of scan rotation $\mathbf{R}$ and translation $\mathbf{t}$ i.e.

$$\frac{\partial \Delta_{u_{i_j}}}{\partial q_0}, \frac{\partial \Delta_{u_{i_j}}}{\partial q_1}, \frac{\partial \Delta_{u_{i_j}}}{\partial q_2}, \frac{\partial \Delta_{u_{i_j}}}{\partial q_3}, \frac{\partial \Delta_{u_{i_j}}}{\partial t_1}, \frac{\partial \Delta_{u_{i_j}}}{\partial t_2}, \frac{\partial \Delta_{u_{i_j}}}{\partial t_3}$$

Using these expression the iteration goes as follows:

$$\begin{aligned} \boldsymbol{\Gamma}_1 &= \boldsymbol{\Gamma}_0 - (\mathbf{J}^{\mathbf{T}}\mathbf{J} + \lambda \mathbf{I})^{-1} \mathbf{J} \boldsymbol{\Delta}_0 && (5.5) \\ \mathbf{J}_{ij} &= \frac{\partial \Delta_i}{\partial \Gamma_j} && (5.6) \\ \boldsymbol{\Gamma} &= \{\{\mathbf{R}\}, \{\mathbf{t}\}\} && (5.7) \end{aligned}$$

Where $\mathbf{J}$ is the Jacobian matrix, $\lambda$ is the Levenberg-Marquardt parameter, $\boldsymbol{\Gamma}_0$ is the initial value at iteration 0 and $\boldsymbol{\Gamma}_1$ is the corrected value after the first iteration. The better convergence of Levenberg-Marquardt method results from an adaptive mixing of Gauss-Newton and Steepest descent step. This mixing depends on the value of $\lambda$. For more details see (Björck, 1996; Press et al., 1988).

The expressions for these partial derivatives will be derived in the following

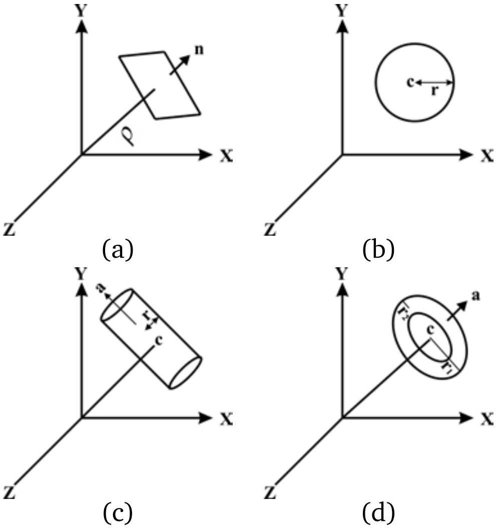


**Figure 5.3:** *Objects used for registration and their parameters (a) Parameters of the plane, normal* $\mathbf{n}$ *and distance from the origin* $\rho$ *(b) Parameters of the sphere, center* $\mathbf{c}$ *and the radius* $r$ *(c) Cylinder is parameterized by axial direction* $\mathbf{a}$*, point closest to the origin* $\mathbf{c}$ *and the radius* $r$*. Note* $\mathbf{a} \perp \mathbf{c}$*. (d) Parameters of the Torus are its center* $c$*, normal direction of its plane* $a$*, major radius* $r_1$ *and minor radius* $r_2$

sections.

### 5.3.1 Spheres

A sphere is defined by its center point $\mathbf{c} = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$ and its radius $r$ (Figure 5.3(a)). Only the center point of the sphere $\mathbf{c}$ is affected by the transformations of its originating scan. This makes a sphere-correspondence analogous to specifying a 3D corresponding point. This combined with the fact that a sphere has optimal fitting accuracy in all directions are the main reasons for the use of spheres of known radii as artificial registration targets.

If a sphere $S_1$ modeled in scan $\Theta_1$, corresponds to the sphere $S_2$ in scan $\Theta_2$, and we want to estimate the rotation $\mathbf{R}_{21} = \mathbf{R}$ and translation $\mathbf{t}_{21} = \mathbf{t}$; using the framework of Equation 5.4 we have to minimize the difference in center points $\mathbf{c}_\delta$.

$$\mathbf{c}_\delta = \mathbf{c}_2 - \mathbf{c}_1 \tag{5.8}$$

$$= \mathbf{R}\mathbf{c}_2 + \mathbf{t} - \mathbf{c}_1 \tag{5.9}$$

The partial derivatives with respect to the registration parameters are as follows:

$$\frac{\partial \mathbf{c}_\delta}{\partial q_i} = \frac{\partial \mathbf{R}}{\partial q_i} c_{1_i} \tag{5.10}$$

$$\frac{\partial \mathbf{c}_\delta}{\partial t_i} = \frac{\partial \mathbf{t}}{\partial t_i} \tag{5.11}$$

Each pair of corresponding spheres will give 3 equations corresponding to 3 components of difference vector $\mathbf{c}_\delta$. Although the radius of the sphere-pair contains no geometrical information about the registration parameters, radius similarity can be used as a constraint for checking the validity of the correspondence especially when spheres with multiple radii are present in the scene i.e.

$$|r_2 - r_1| \leq r_\delta \tag{5.12}$$

Where $r_\delta$ is a threshold

### 5.3.2 Planes

In industrial environments many objects consist of one or more planar faces that can be used for registration. Describing a plane by the normal vector $\mathbf{n} = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}^T$ and the perpendicular distance from the origin $\rho$ (Figure 5.3(b)) provides a singularity free representation (Heuvel, 1999). This representation is also known as Hesse form of the plane. As there can be only three degrees of freedom for a plane, we have a constraint on the length of the normal vector $\mathbf{n}$

$$\|\mathbf{n}\| = \sqrt{n_x^2 + n_y^2 + n_z^2} = 1 \tag{5.13}$$

The given formulation is for infinite planes. As point cloud data is usually very noisy on edges, accurate determination of bounding curves for planar faces is quite difficult, and we have to use infinite planes.

For the indirect method each plane provides three equations for the difference in normal vector $\mathbf{n}_\delta$ and one equation for the difference in the distance from the origin $\rho_\delta$ which are given by:

$$\mathbf{n}_\delta = \mathbf{n}_1 - (\mathbf{R}\mathbf{n}_2) \tag{5.14}$$

$$\rho_\delta = \rho_2 - \rho_1 + (\mathbf{R}\mathbf{n}_2) \cdot \mathbf{t} \tag{5.15}$$

From Equation 5.14 we see that the normal is only affected by the rotation, whereas Equation 5.15 shows that the change in $\rho$ is a function of both translation and rotation of the scan. The partial derivatives w.r.t rotation are as follows:

$$\frac{\partial \mathbf{n}_\delta}{\partial q_i} = -\frac{\partial \mathbf{R}}{\partial q_i}\mathbf{n}_2 \tag{5.16}$$

$$\frac{\partial \rho_\delta}{\partial q_i} = (\frac{\partial \mathbf{R}}{\partial q_i}\mathbf{n}_2) \cdot \mathbf{t} \tag{5.17}$$

For translation $\mathbf{t}$ we have the following expressions:

$$\frac{\partial \mathbf{n}_\delta}{\partial t_i} = 0 \tag{5.18}$$

$$\frac{\partial \rho_\delta}{\partial t_i} = (\mathbf{R}\mathbf{n}_2) \cdot \frac{\partial \mathbf{t}}{\partial t_i} \tag{5.19}$$

### 5.3.3 Cylinder

Cylinders are quite frequently encountered on industrial sites, either as pipes or as parts of more complex objects. It can be quite difficult to accurately determine the start and end point of cylinders from point clouds in many cases due to occlusions and point spacing. For this reason we employ a parametrization for the infinite cylinder.

We represent a cylinder by 7 parameters, 3 for the axis of cylinder $\mathbf{a}$, 3 for the point closest to the origin $\mathbf{c}$ and one for radius $r$ (Figure 5.3(c)). As there are only 5 degrees of freedom for cylinder we have two constraints:

$$\|\mathbf{a}\| = 1 \tag{5.20}$$

$$\mathbf{a} \perp \mathbf{c} \Rightarrow \mathbf{a} \cdot \mathbf{c} = 0 \tag{5.21}$$

As the use of unbounded objects increases the degrees of freedom, we need more correspondences to calculate the registration parameters. For example one correspondence pair of a bounded cylinders is enough to calculate the registration up to a cylindrical rotation, while for unbounded cylinders one correspondence leaves a cylindrical rotation along with a linear shift undetermined. The condition of non-degeneracy is important because in a degenerate case the solution can either be numerically ill-determined or some of the parameters can be completely undetermined. For example correspondences of multiple parallel cylinders do not provide enough information for the determination of translation along their axial direction.

Each cylinder correspondence gives us 6 equations but only 4 of them are independent. Radius for cylinders behaves in a manner similar to that of spheres (Section 5.3.1).

The axial direction of cylinder $\mathbf{a}$ is similar to normal direction for plane $\mathbf{a}$, and its partial derivatives have exactly the same expressions. (Section 5.3.2).

The point closest to origin depends on both translation and rotation of the scan the cylinder comes from. Additionally the constraint $\mathbf{a} \perp \mathbf{c}$ requires that we subtract the projection of $\mathbf{c}$ along cylinder axis $\mathbf{a}$. If a cylinder $C_1$ from scan $\Theta_1$ corresponds to the cylinder $C_2$ from scan $\Theta_2$, the difference in their closest points after transformation of the second scan are given by:

$$\mathbf{c}_\delta = -\mathbf{c}_1 + (\mathbf{R}\mathbf{c}_2 + \mathbf{t} - (\mathbf{R}\mathbf{n}_2 \cdot \mathbf{t})(\mathbf{R}\mathbf{n}_2)) \tag{5.22}$$

$$= -\mathbf{c}_1 + (\mathbf{R}\mathbf{c}_2 + \mathbf{t} - (\mathbf{n}_R \cdot \mathbf{t})\mathbf{n}_R) \tag{5.23}$$

Where $\mathbf{n}_R = \mathbf{R}\mathbf{n}_2$ Its partial derivatives w.r.t rotation and translation parameters are as follows:

$$\frac{\partial \mathbf{c}_\delta}{\partial q_i} = \frac{\partial \mathbf{R}}{\partial q_i}\mathbf{c}_2 - ((\mathbf{n}_R \cdot \mathbf{t})\frac{\partial \mathbf{n}_R}{\partial q_i} + (\frac{\partial \mathbf{n}_R}{\partial q_i} \cdot \mathbf{t})\mathbf{n}_R) \tag{5.24}$$

$$\frac{\partial \mathbf{c}_\delta}{\partial t_i} = \frac{\partial \mathbf{t}}{\partial t_i} - \frac{\partial(\mathbf{n}_R \cdot \mathbf{t})}{\partial t_i}\mathbf{n}_R \tag{5.25}$$

$$= \frac{\partial \mathbf{t}}{\partial t_i} - n_{R_i}\mathbf{n}_R \tag{5.26}$$

Where $\frac{\partial \mathbf{n}_R}{\partial q_i} = \frac{\partial \mathbf{R}}{\partial q_i}\mathbf{n}_2$.

### 5.3.4 Torus

A full torus is seldom used for the design of industrial sites, but partial tori especially in the form of bends and elbows between orthogonal pipes are quite common. Similar to the case of cylinders (Section 5.3.3) we would not use bounded tori as it is quite difficult to estimate the start and end-angle accurately from point clouds.

A torus is given by 8 parameters, 3 for the axis $\mathbf{a}$, 3 for the central point $\mathbf{c}$, two for the radii; $r1$ for the outer radius and $r2$ for the radius of the cross section (Figure 5.3(d)). There is one constraint namely $\|\mathbf{a}\| = 1$.

We do not need to formulate any separate equations for torus, and its axis direction can be treated similar to the normal of the plane (Section 5.3.2). Similarly, the center point of the torus can be treated similar to that of the sphere (Section 5.3.1). The radii can be used for checking the validity of correspondences but they do not have any useful geometric information for registration for the Indirect method.

## 5.4 Direct Method

As it was noted in the introduction the Indirect Method is geometrically approximate. The second method, called *Direct method*, takes the geometrically exact approach. In this method the sum of the squares of the orthogonal distance of points from their model surfaces are minimized with respect to the registration parameters of the scan and the shape parameters of the object the points come from.

The direct method can be slower than the indirect method as the number of observations depends on the number of points which can be in the order of millions for most industrial modeling projects. In contrast the number of observations in Indirect method was just a function of the objects modeled

Additionally, the convergence of the direct method requires good approximate values which can be obtained from the indirect method. This way both methods complement each other, and can be used in a serial fashion.

Direct Method consists of two steps:

1. Registration Estimation, resulting in estimation of only the registration parameters of the scan
2. Integrated Adjustment, where both registration parameters and shape parameters are simultaneously estimated.

The first step of *Registration estimation* is necessary to refine the approximate values coming from the Indirect method. Based on this better estimate of the transformation more corresponding object pairs are automatically added to the correspondence list. This augmented list is then used during *Integrated adjustment* to estimate both scan transformation parameters and object shape parameters.

The merit function for the first step can be given as:

$$\min_{\{\mathbf{R}\},\{\mathbf{t}\}} \sum_{i=1}^{o} \sum_{j=1}^{p^i} \boldsymbol{\Omega}_i^2(\mathbf{R}_i \mathbf{p}_j + \mathbf{t}_i) \tag{5.27}$$

In Equation 5.27, $o$ is the total number of objects, and $p^i$ are the number of points from all un-registered point clouds which have been used to model $i-$th object. Function $\boldsymbol{\Omega}_i$ calculates the orthogonal distance of a given point from $i-$th object's surface. We want to minimize $\boldsymbol{\Omega}_i^2$, by changing the transformation parameters $\{\mathbf{R}\}$ and $\{\mathbf{t}\}$.

The second step in the process of object based registration extends the idea of *Bundle adjustment* from traditional Photogrammetry (Atkinson, 1996). In this step we minimize the sum of squares of orthogonal distance of the points from their respective model surfaces with respect to shape and pose parameters of the object in addition to the registration parameters of the scan. We name this stage *Integrated Adjustment*, as it integrates the estimation of all unknowns together. As we show in the final section this leads to better estimation accuracies and does not result in the propagation of registration errors to the modeling stage, instead it combines both steps together and provides simultaneous use of the available information.

$$\min_{\{\mathbf{R}\},\{\mathbf{t}\},\{\mathbf{u}\}} \sum_{i=1}^{o} \sum_{j=1}^{p^i} \boldsymbol{\Omega}_i^2(\mathbf{R}_i \mathbf{p}_j + \mathbf{t}_i, \mathbf{u}_i) \tag{5.28}$$

In Equation 5.28 all variables are similar to Equation 5.27, except now we add the model parameters ($\{\mathbf{u}\}$) to the set of minimization parameters.

Now we will derive partial derivatives required for both stages of the *Direct Method*.

In the following discussion most expressions will be derived in terms of a measured point $\mathbf{p}$:

$$\mathbf{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix} \tag{5.29}$$

### 5.4.1 Plane

Using the parametrization from Section 5.3.2 the distance $d$ of a transformed point $\mathbf{v}_p$ from the plane is given as:

$$\begin{aligned} \mathbf{v}_p &= \mathbf{R}\mathbf{p} + \mathbf{t} & (5.30) \\ d &= \mathbf{v}_p \cdot \mathbf{n} - \rho & (5.31) \\ &= v_{p_x} n_x + v_{p_y} n_y + v_{p_z} n_z - \rho & (5.32) \end{aligned}$$

Here we will derive partial derivatives for $\mathbf{v}_p$ only, the full expressions for $\frac{\partial d}{\partial x}$ can be evaluated by using the chain rule.

$$\begin{aligned} \frac{\partial \mathbf{v}_p}{\partial q_i} &= \frac{\partial \mathbf{R}}{\partial q_i}\mathbf{p} & (5.33) \\ \frac{\partial \mathbf{v}_p}{\partial t_i} &= \frac{\partial \mathbf{t}}{\partial t_i} & (5.34) \end{aligned}$$

For plane parameters we have to use full expression for $d$ (Equation 5.32) but the resulting expressions are as follows:

$$\begin{aligned} \frac{\partial d}{\partial n_i} &= v_{p_i} & (5.35) \\ \frac{\partial d}{\partial \rho} &= -1 & (5.36) \end{aligned}$$

### 5.4.2 Sphere

We use the parametrization of sphere given in Section 5.3.1. As for the case of plane we will define a subexpression $\mathbf{v}_s$ that will be used for simplifying the expressions.

$$\mathbf{v}_s = (\mathbf{R}\mathbf{p} + \mathbf{t}) - \mathbf{c} \qquad (5.37)$$

The distance $d$ of a transformed point from this sphere can be expressed as:

$$\begin{aligned} d &= \|\mathbf{v}_s\| - r & (5.38) \\ &= \sqrt{\mathbf{v}_s \cdot \mathbf{v}_s} - r & (5.39) \\ &= \sqrt{v_{s_x} v_{s_x} + v_{s_y} v_{s_y} + v_{s_z} v_{s_z}} - r & (5.40) \end{aligned}$$

The expressions for $\frac{\partial \mathbf{v}_s}{\partial q_i}$ and $\frac{\partial \mathbf{v}_s}{\partial t_i}$ are same as $\frac{\partial \mathbf{v}_p}{\partial q_i}$ and $\frac{\partial \mathbf{v}_p}{\partial t_i}$ respectively (Section 5.4.1).

For $\frac{\partial \mathbf{v}_s}{\partial c_i}$ the expressions are:

$$\frac{\partial \mathbf{v}_s}{\partial c_i} = -c_i \tag{5.41}$$

The chain rule can be used for calculating $\frac{\partial d}{\partial x}$ where $x$ is any parameter:

$$\frac{\partial d}{\partial x} = \frac{1}{\|\mathbf{v}_s\|}(v_{s_x}\frac{\partial v_{s_x}}{\partial x} + v_{s_y}\frac{\partial v_{s_y}}{\partial x} + v_{s_z}\frac{\partial v_{s_z}}{\partial x}) - \frac{\partial r}{\partial x} \tag{5.42}$$

### 5.4.3 Cylinder

The distance of a transformed point from the cylinder is given by:

$$\begin{aligned} d &= \|(\mathbf{Rp} + \mathbf{t} - \mathbf{c}) \times \mathbf{a}\| - r & (5.43)\\ &= \|\mathbf{v}_c\| - r & (5.44)\end{aligned}$$

Where $\mathbf{v}_c$ is defined as:

$$\mathbf{v}_c = (\mathbf{Rp} + \mathbf{t} - \mathbf{c}) \times \mathbf{a} \tag{5.45}$$

Using the definition of $\mathbf{v}_s$ from Section 5.4.2, we can rewrite it as:

$$\begin{aligned} \mathbf{v}_c &= \mathbf{v}_s \times \mathbf{a} & (5.46)\\ \frac{\partial \mathbf{v}_c}{\partial x} &= \frac{\partial}{\partial x}\begin{bmatrix} a_z v_{s_y} - a_y v_{s_z} \\ a_x v_{s_z} - a_z v_{s_x} \\ a_y v_{s_x} - a_x v_{s_y} \end{bmatrix} & (5.47)\end{aligned}$$

Using Equation 5.47, with expressions $\frac{\partial \mathbf{v}_s}{\partial x}$ from Section 5.4.2, and applying chain rule $\frac{\partial \mathbf{v}_c}{\partial x}$ can be calculated.

### 5.4.4 Torus

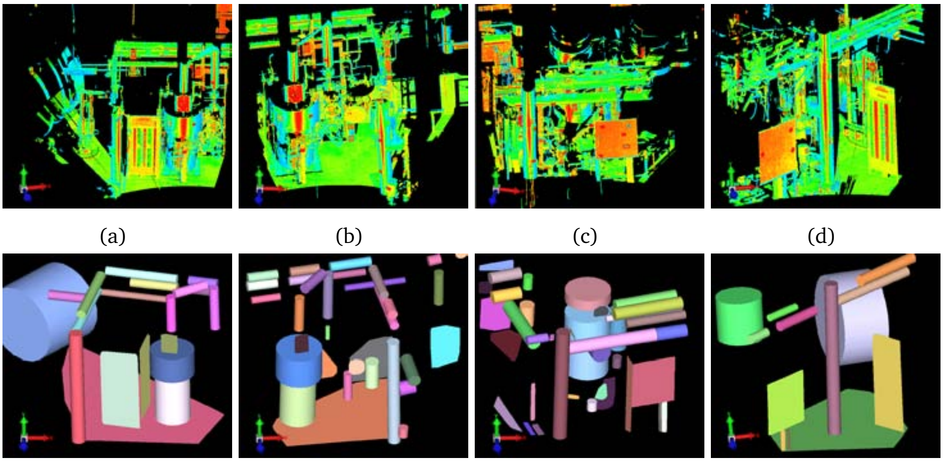
The parametrization of the torus used here is the same as the one given in Section 5.3.4. To calculate the distance of a transformed point from the torus surface we define two auxiliary variables $\mathbf{v}_{tl}$ that is the distance from the line passing through the center, and $\mathbf{v}_{tp}$ distance of the point from the plane passing through the center and with normal equal to $\mathbf{a}$.

$$\begin{aligned} \mathbf{v}_{tl} &= (\mathbf{Rp} + \mathbf{t} - \mathbf{c}) \times \mathbf{a} & (5.48)\\ \mathbf{v}_{tp} &= (\mathbf{Rp} + \mathbf{t} - \mathbf{c}) \cdot \mathbf{a} & (5.49)\end{aligned}$$

Using these definitions the distance $d$ of a point from the torus surface can be formulated as:

$$d = \sqrt{(\|\mathbf{v}_{tl}\| - r_1)^2 + \mathbf{v}_{tp} \cdot \mathbf{v}_{tp}} - r2 \tag{5.50}$$

Now $\mathbf{v}_{tl}$ can be treated similar to cylindrical $\mathbf{v}_c$ from Section 5.4.3,while $\mathbf{v}_{tp}$ can be treated as planar $\mathbf{v}_p$ from Section 5.4.1. Those expressions along with application


(a) (b) (c) (d)

(e) (f) (g) (h)

**Figure 5.4:** *(a–d) Scan 1,2,3, and 4 used for registration experiment (e–h) objects modeled from the scans and used for registration through Indirect and Direct methods*

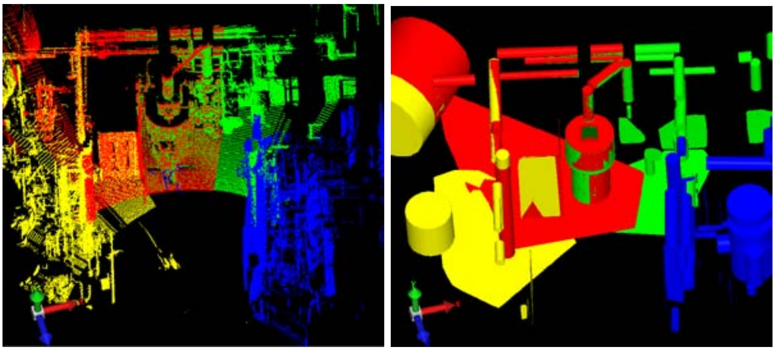
of chain rule gives us the partial derivatives for torus.

## 5.5 Results and discussion

We implemented the presented methods of Indirect and Direct registration in C++. The results of their application to some point clouds from an industrial site are presented in this section. These point clouds were acquired using a Cyrax HDS2500 (Leica HDS, 2005) laser scanner. Each scan consisted of one million points, and four scans were used for the presented experiment. The individual scans are shown in Figure 5.4(a–d).

We will compare the presented methods with industry standard implementations of ICP and show that our approach performs better in two aspects. Firstly, the presented methods do not need placement of artificial targets for calculation of approximate values, which is needed most of the time for ICP. Secondly, our approach combines both registration and fitting in one estimation procedure and as a result the final fitting error as measured by the average residual and standard deviation of parameter estimates is less for our approach compared to that of the ICP.

First we will give details of the presented methods, and the comparison with ICP will follow. The scans were segmented and planes and cylinders were semi-automatically recovered using methods presented in (Rabbani and Heuvel, 2004a; Vosselman et al., 2004). The fitted models are shown in Figure 5.4(e–h). The



(a) (b)

**Figure 5.5:** *Results of Indirect method of registration (a) Combined transformed point cloud (b) Combined transformed objects. Scan 1–4 are shown in red, green, blue and yellow colors respectively.*

number of recovered objects in scan 1–4 were 28, 44, 46 and 19 respectively.

After modeling the scans were registered pair-wise using the Indirect method. As pointed out earlier, this is a quick way to get approximate values for the second step of the Direct registration. This step of Indirect method was applied for pair-wise registration. Although theoretically it is possible to do simultaneous registration of multiple scans good approximate values can be obtained from pair-wise registration. The coordinate system of scan 1 was defined as the world coordinate system. The pair wise registration parameters of the rest of the scans were calculated through appropriate transformation concatenations. In contrast to ICP no artificial targets were used for this step.

The results of this step are shown in Figure 5.5, where visually the registration looks acceptable as approximate values for the next step. The remaining errors in the registration can be visually seen for example in the big cylinders of scan 1 and 2, which still do not fit perfectly. The values of estimated parameters are shown in Table 5.1.

Although the correspondence specification in the step of Indirect registration is done manually this process can be automated using either exhaustive search or RANSAC (Fischler and Bolles, 1987) based random search through the parameter space. In case of only planes and spheres a minimum of three correspondences are required for registration, whereas for cylinders and tori this number is two.

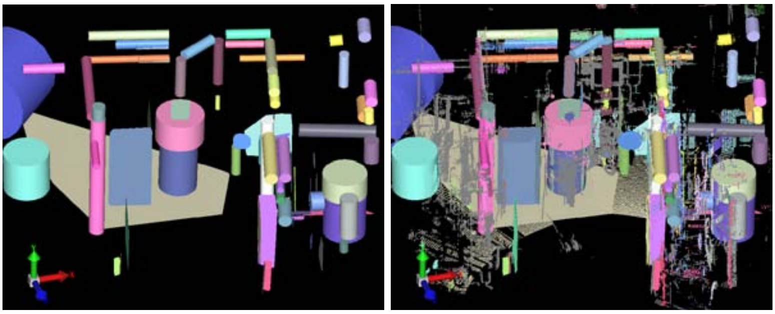
Before proceeding with the next stage of the direct method, the corresponding objects must be combined so that they can be used for estimating the registration parameters of the scans along with their own shape parameters. This process is automated by doing a distance based search, and all objects within a certain distance from one model are combined together. This step also combines multiple objects in one scan which are close together. For example if there are multiple planar patches in one scan that can be explained by one bigger planar patch, they

**Table 5.1:** *Registration parameters from Indirect method*

| Scan | $q_0$ | $q_1$ | $q_2$ | $q_3$ | $t_x$ (m) | $t_y$ (m) | $t_z$ (m) |
|---|---|---|---|---|---|---|---|
| 2 | 0.9776 | 0.00023 | -0.2046 | -0.009 | 0.0041 | 0.0069 | 0.0228 |
| 3 | -0.4992 | -0.0082 | 0.8738 | 0.0112 | -1.0868 | 0.1435 | -3.735 |
| 4 | 0.9549 | 0.0123 | 0.2964 | 0.0224 | 0.9942 | -0.0084 | 0.5436 |

**Table 5.2:** *Registration parameters from Direct method*

| Scan | $q_0$ | $q_1$ | $q_2$ | $q_3$ | $t_x$ (m) | $t_y$ (m) | $t_z$ (m) |
|---|---|---|---|---|---|---|---|
| 2 | 0.9776 | 0.00040 | -0.2101 | -0.0045 | -0.0475 | -0.0199 | 0.0108 |
| 3 | -0.4861 | -0.0078 | 0.8738 | 0.0084 | -1.0855 | 0.1183 | -3.785 |
| 4 | 0.9549 | 0.0109 | 0.2958 | 0.0229 | 0.9851 | 0.0086 | 0.5411 |



(a) (b)

**Figure 5.6:** *Results of combining objects after Indirect method, for proceeding with the Direct method. Note the objects in the overlap area are merged and given one color (a) Combined objects (b) Objects with points*

are combined into one object. At the end of this combining step for the given example there are 2 objects which have points from three scans, 14 objects have points from two scans and the rest of 72 objects have points from only one scan. The last category of objects does not contribute any information for improving the registration parameters. But by including them in Direct method, we make sure that any changes in the registration from the overlapping objects is also reflected in the parameters of the non-overlapping ones.

For the direct method the number of the parameters and the observations can be very large and this can slow down the estimation process. In the presented example we had 88 objects, and four scans. The registration parameters of scan 1 were kept fixed to define the world coordinate system. In total we had to estimate 538 parameters. Although all all parameters were estimated, we show only the ones for scan registration in Table 5.2.

At the start of the Direct method the rms error using the approximate solution from

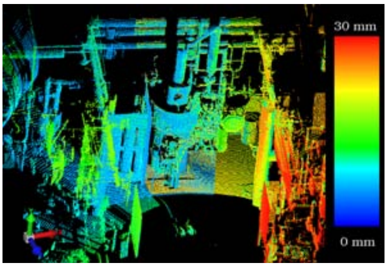


**Figure 5.7:** *Distance of the point cloud resulting from Indirect method from the one from Direct method*

**Table 5.3:** *Standard deviations from Direct method*

| Scan | $q_0$ | $q_1$ | $q_2$ | $q_3$ | $t_x$ (m) | $t_y$ (m) | $t_z$ (m) |
|---|---|---|---|---|---|---|---|
| 2 | 3.221e-5 | 2.118e-5 | 1.306e-5 | 4.363e-5 | 1.607e-4 | 2.491e-4 | 7.984e-5 |
| 3 | 6.105e-4 | 2.835e-4 | 3.356e-4 | 6.730e-3 | 2.266e-3 | 1.609e-3 | 2.283e-3 |
| 4 | 3.003e-4 | 1.893e-4 | 1.627e-4 | 5.030e-3 | 1.576e-3 | 4.435e-4 | 1.031e-3 |

Indirect method was 5.12 mm. After the Direct method this was reduced to 4.09 mm. This is the rms value of distances of the points from their originating models. Ideally this error should go down to zero but even in man-made environments the objects only approximately conform to mathematical models. For example not many objects modeled as planes are perfectly planar. Additionally, there can be big discrepancies like bent or squashed pipes, pipe insulations etc. These errors combined with the measurement noise of laser scanner explain this non-zero total squared error. Furthermore, as the values are less than the stated measurement noise we can assume there are no serious deformations present.

To evaluate the relative improvements obtained by the Direct method we calculated the distance of the point cloud resulting from the Direct method from the one from Indirect method. This distance (calculated as the closest point in the result of Direct method to the one from Indirect method) is shown as a color coded point cloud in Figure 5.7. There we see the biggest difference in the points of scan 3. This is because there is very little overlap in terms of objects, and the approximation of Indirect method for this scan was more erroneous compared to the other scans where we had comparatively better overlap.

The standard deviations of the transformation parameters were obtained by propagating the standard deviations of the point measurements to the object parameters. For the point cloud acquisition the covariance matrix $5^2\mathbf{I}_n$ was chosen based on the data provided by scanner manufacturer (Leica HDS, 2005), which states a standard deviation of $5$ mm. The resulting standard deviations for the scan registration parameters are shown in Table 5.3. Again we see the higher values for scan 3, explained by comparably lower overlap.

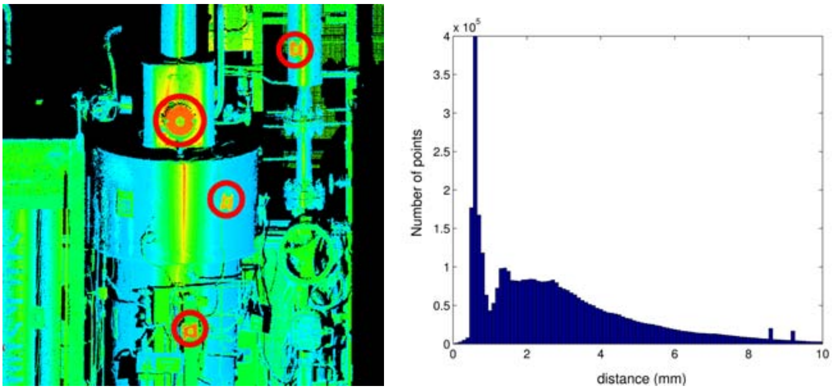


**Figure 5.8:** *(a)Targets used for ICP-based registration in Cyclone (b) histogram of distance between the result of Direct method and ICP.*

Compared to ICP this is quite a unique and useful feature of the presented method. By providing a full covariance matrix for registration parameters of the scan and the shape parameters of the object, the quality of the results can be judged mathematically. This also provides a way to identify the cases where more measurements/scans should be added for improvement of the results. The standard deviation values for translations have a direct interpretation, but for quaternions the conversion from quaternions to the standard Euler angles (Shoemake, 1985) is useful. For example for Scan 2 the standard deviations in Euler angles are $\begin{pmatrix} 0.0033 & 0.0023 & 0.0044 \end{pmatrix}$ degrees.

To register the same set of point clouds using ICP artificial targets were placed to get enough corresponding points for initial values. Some of these targets are shown in the Figure 5.8(a). This highlights one of the main strengths of the presented approach as for industrial and man-made environments, like buildings etc, our method uses the available geometric information for the calculation of approximate values and thus does not require artificial targets. For example in the presented scenario the geometric information from corresponding planes and cylinders was sufficient for initial value calculation.

For the registration through ICP point cloud processing software from Leica Geo System called *Cyclone* was used (Leica HDS, 2005). For comparing the registration from Direct method with the results of ICP we first calculated the distance of the closest points between the resultant point clouds of the two methods. A histogram of these distances is shown in Figure 5.8(b). The mean distance was 2.9 mm while the standard deviation was 2.1 mm.

As we said in the introduction the traditional approach leads to accumulation of errors as registration and modeling are two separate steps without any error propagation. Moreover, the implementation of ICP in Cyclone as well as in most commercial modeling programs can handle only pair-wise registration, which leads to non optimal use of the available information.

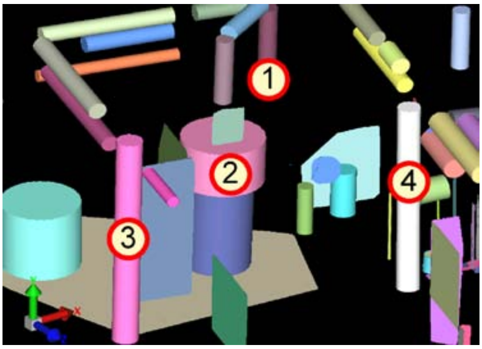


**Figure 5.9:** *Cylinders used for comparing ICP with Direct method. (Table 5.4 & 5.5)*

**Table 5.4:** *Comparison of ICP and Direct method. Parameter values and average residuals*

| No. | Method | $a_x$ | $a_y$ | $a_z$ | $t_x$ (m) | $t_y$ (m) | $t_z$ (m) | $r$ (m) | avg. res ($m^2$) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Direct | -0.0026 | 0.9984 | 0.0561 | 0.7754 | 0.2423 | -4.2784 | 0.0949 | 6.06e-06 |
| | ICP | -0.0052 | 0.9984 | 0.0563 | 0.7755 | 0.2452 | -4.2787 | 0.0947 | 9.56e-06 |
| 2 | Direct | 0.0021 | 0.9991 | 0.0431 | 0.7802 | 0.1822 | -4.2601 | 0.4462 | 2.18e-05 |
| | ICP | -0.0008 | 0.9991 | 0.0432 | 0.7804 | 0.1849 | -4.2608 | 0.4461 | 2.26e-05 |
| 3 | Direct | -0.0131 | 0.9990 | 0.0436 | -1.0600 | 0.1282 | -3.2529 | 0.1227 | 1.93e-05 |
| | ICP | -0.0121 | 0.9990 | 0.0429 | -1.0574 | 0.1268 | -3.2526 | 0.1214 | 2.11e-05 |
| 4 | Direct | -0.0195 | 0.9992 | 0.0343 | 1.9565 | 0.1265 | -2.5709 | 0.1247 | 8.95e-06 |
| | ICP | -0.0244 | 0.9991 | 0.0346 | 1.9538 | 0.1368 | -2.5697 | 0.1219 | 1.01e-05 |

**Table 5.5:** *Comparison of ICP and Direct method. Parameter standard deviations.*

| No. | Method | | $a_x$ | $a_y$ | $a_z$ | $t_x$ (m) | $t_y$ (m) | $t_z$ (m) | $r$ (m) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Direct | $\times 10^{-3}$ | 0.3780 | 0.0130 | 0.3445 | 0.0570 | 0.0075 | 0.1325 | 0.1095 |
| | ICP | $\times 10^{-3}$ | 0.4005 | 0.0160 | 0.3865 | 0.0590 | 0.0080 | 0.1365 | 0.1135 |
| 2 | Direct | $\times 10^{-3}$ | 0.2140 | 0.0067 | 0.1750 | 0.0365 | 0.0045 | 0.1075 | 0.0835 |
| | ICP | $\times 10^{-3}$ | 0.2270 | 0.0070 | 0.1665 | 0.0390 | 0.0045 | 0.1050 | 0.0850 |
| 3 | Direct | $\times 10^{-4}$ | 0.360 | 0.0150 | 0.3000 | 0.4450 | 0.0250 | 0.6750 | 0.6300 |
| | ICP | $\times 10^{-4}$ | 0.3850 | 0.0150 | 0.340 | 0.5300 | 0.0250 | 0.7250 | 0.7200 |
| 4 | Direct | $\times 10^{-4}$ | 0.2750 | 0.0150 | 0.3400 | 0.4650 | 0.0150 | 0.2050 | 0.3700 |
| | ICP | $\times 10^{-4}$ | 0.3200 | 0.0150 | 0.3850 | 0.5250 | 0.0100 | 0.2250 | 0.4000 |

We compared the results of the Direct method with those of ICP for accuracy of model estimation. For this purpose four cylinders in different areas of the scans were selected (Figure 5.9). Cylinders were fitted to the resultant point cloud from ICP and the one from the Direct method, and the fitting results were compared based on the standard deviations of the estimated parameters as well as the average residual error. The results of this comparison are given in Table (5.4 & 5.5). For the case of all four cylinders the average residual decreased for our approach, in some cases up to a factor of 1.5. Similarly the standard deviation of the estimated parameters is in general lower for our approach. This shows the importance of simultaneous registration as well as the integrated estimation of registration and model parameters.

As noted before, our approach has one serious assumption regarding the conformance of the captured data to the mathematical models used for fitting. Although in most cases the difference between an ideal model like a cylinder and the data captured from say a piping installation can be considered small, in some cases we can expect big discrepancies. In such cases the presented approach can run into problems and can only give an approximate solution. In general a model accuracy based comparison, similar to the one we did in this section, is a good indicator for how far the model conformance assumption is being violated. In such situations if there is sufficient overlap, this solution can be improved upon by ICP. Whether such improvement can actually be obtained can be checked by the increase/decrease in the standard deviation of parameter estimates for models. But using our approach, we still get the benefits of good approximate values without using artificial targets, which can lead to a big reduction in time and manual labor required. Additionally, the potential for the automation of the correspondence search can further decrease the amount of manual input required.

## 5.6 Conclusions

Two methods for registration of the point clouds taken from different viewpoints have been presented. Both methods make use of automatic/manual segmentation that labels the points as belonging to a certain object. Both strategies can be used for the simultaneous registration of multiple point clouds. The Indirect method, though geometrically flawed, provides a quick way to get approximate values based on manual specification of correspondences. These approximate values are latter used by the Direct method.

The Direct method extends the idea of bundle adjustment from traditional photogrammetry. It estimates the registration parameters of the scan and the shape parameters of the objects by minimizing the orthogonal distances of the points from the object surface. This is a nonlinear estimation process and the partial derivatives required for its implementation were derived.

The presented methods were applied to a test data-set from an industrial site, and although artificial targets were required for ICP, our method was able to register the point clouds using only the modeled objects. Additionally, quality estimates in terms of standard deviations of all scan and object parameters were computed. The presented approach performs simultaneous registration and thus avoids accumulation of errors resulting from pair-wise registration, using all available information for every step of the process. We compared the presented approach with the industry standard implementations of ICP and showed that our approach performed better in terms of achieving a lower average residual for model fitting, and giving a better estimation accuracy of model parameters.

In the next chapter we will present efficient strategies for implementing automatic search for finding corresponding objects for model based registration.

# Chapter 6

# Automatic correspondence search

> Study the past if you would define the future.
>
> Confucius

> Look to the past for guidance into the future.
>
> Robert Jacob Goodkin

> Those who cannot remember the past are condemned to repeat it.
>
> George Santayana

In this chapter we present a set of constraints that can be used to efficiently search for corresponding objects in a given set of scans. These constraints result in a considerable reduction in the computational complexity of the required combinatorial search for corresponding objects. Depending on the type of the object there is a minimum number of correspondence pairs which are necessary to determine the complete rigid transformation. Each match provides partial information towards the calculation of this transformation which can be interpreted as a constraint. These constraints can be used to reduce the number of possibilities for each new match based on the information from the previous matches. In this chapter we give a mathematical formulation of such constraints and details of their implementation. Once a minimal set of corresponding objects have been found, the scans are approximately aligned, and can be searched for other overlapping objects. These corresponding objects can be used in an *Integrated adjustment* (Chapter 5) for final registration. We compare the performance of RANSAC based correspondence search using the presented constraints with the blind RANSAC search, which shows that the presented method leads to significant improvements in reducing the computational complexity of the problem.

## 6.1 Introduction

As noted in the last chapter most of the commonly used registration algorithms make minimal assumptions about the types of objects present in the scans. This makes them very general-purpose but at the same the domain specific information that could have been used for automation is thrown away. We have taken the opposite approach as we restrict our domain to the industrial scenarios and we assume that well-defined objects are present in the point-clouds, which can be used as targets for registration. In the last chapter the corresponding objects were specified manually, but we mentioned that this search can be automated by solving the problem of finding best set of corresponding objects. In this chapter we address how to efficiently implement such a search.

The correspondence search can be implemented in two ways: either by using a tree search that explores all/selected possibilities using different tree search methods like depth-first, breadth-first and A* (Russell and Norvig, 2003; Vosselman, 1995). Alternatively we can use a randomized search for example based on RANSAC (Fischler and Bolles, 1987). As we have different number of minimum matches required for each type of objects for the determination of complete rigid body transformation (e.g. 3 corresponding planes or 2 corresponding cylinders), we have a sequential process where one match is selected followed by the next one. Each match provides partial information towards the final solution of the registration problem. This partial information can be interpreted as a constraint that narrows down the number of options for the next match. These constraints allow us to use the available information fully at every step, and allows the future decisions to be guided by ones made in the past.

### 6.1.1 Outline

The rest of the chapter is organized as follows. We give an outline of our algorithm in Section 6.2. In Section 6.3 we give the details of geometric constraint propagation that can be used to implement an efficient search for the corresponding objects. Section 6.4 provides a comparison of two search methods using the presented constraints with the ones not using them; both RANSAC and tree search methods are compared. We present some results on industrial point clouds in Section 6.5. Finally we conclude in Section 6.6 with some directions for the future work.

# 6.2 Algorithm outline

## 6.2.1 Naming conventions

For referring to scans and corresponding objects we will follow the following notation. The correspondence search is presented for a pair of scans, the one being used as the reference is called $\mathbf{\Theta}_f$ ( $f$ is for fixed ) and the one being registered is called $\mathbf{\Theta}_r$. The $n$-th match of corresponding objects in the two scans will be labeled as $O_{fn}$ and $O_{rn}$, where $O$ can be $P$, $C$, $S$ or $T$ for the case of a plane, cylinder, sphere and torus respectively. For transformations we will use capital $\mathbf{T}_n^O$, where the subscripts that its the transformation for $n$-th step and for type of object $O$.

## 6.2.2 Algorithm steps

The overall process of object based automatic registration consists of the following steps:

1. Given two scans $\mathbf{\Theta}_f$ and $\mathbf{\Theta}_r$, detect the set of points that belong to simple objects like planes, cylinder and tori automatically using for example segmentation (Chapter 2) and Hough transform (Chapter 3).
2. Fit models to each detected point-set above using least-squares both in $\mathbf{\Theta}_f$ and $\mathbf{\Theta}_r$ (Chapter 4).
3. Find minimal set of object correspondences in $\mathbf{\Theta}_r$ and $\mathbf{\Theta}_f$ and through them the approximate transformation $\mathbf{T}_{approx}$ using the techniques of Section 6.3.
4. The approximate registration of the previous step are used to search for more corresponding objects based on proximity and similarity of geometry.
5. Keep the object parameters fixed, and adjust only for scan parameters using *Direct method* (Chapter 5). The adjustment would minimize the square of orthogonal distance of the points to corresponding object surfaces.
6. Do simultaneous adjustment for both scan and object parameters using techniques of *Integrated adjustment* (Chapter 5).

# 6.3 Constrained Search for Correspondences

The search for finding the best set of corresponding objects is a combinatorial optimization problem. The simplest solution is through an exhaustive search, but this soon becomes computationally infeasible. For example, if the two scans, $\mathbf{\Theta}_r$ and $\mathbf{\Theta}_f$, contain $n_r$ and $n_f$ models and the minimum number of matches required is $m$, the number of possibilities for correspondences is ${}_{n_r}P_m \times_{n_f} P_m$

where ${}_nP_k = \frac{n!}{(n-k)!}$. For larger $n$'s and smaller $m$ the number of possibilities can be approximated by $\approx {n_r}^m \times {n_f}^m$.

Although different search tree optimization methods exist in Artificial Intelligence literature (Russell and Norvig, 2003) and they have been successfully applied in Photogrammetric applications (Vosselman, 1995), most of them (e.g. A*) require a heuristic function that gives the approximate cost function given the current state of the search. This can be quite difficult to provide for the case of the point cloud registration problem, as due to multiple local minima the current state of the search does not provide enough information for calculation of an approximate distance from the final goal. As a result we have selected the technique of constraint propagation for pruning the search tree.

In this pruning approach we apply the geometric constraints resulting from the previous matches to narrow down the possibilities for the current match. Additionally, instead of trying to reach the global minimum we assume a certain overlap between the set of scans being registered (the overlap can be either in terms of objects or that of points) and as soon as we successfully achieve that overlap (as measured by a given distance criterion) the search process is stopped.

The presented constraints are quite general and can be used by both tree-based search and RANSAC like random methods. We will show the improvements in search efficiency using the presented constraints in Section 6.4. Details of these constraints for correspondences of different types of objects are provided in the following sections.

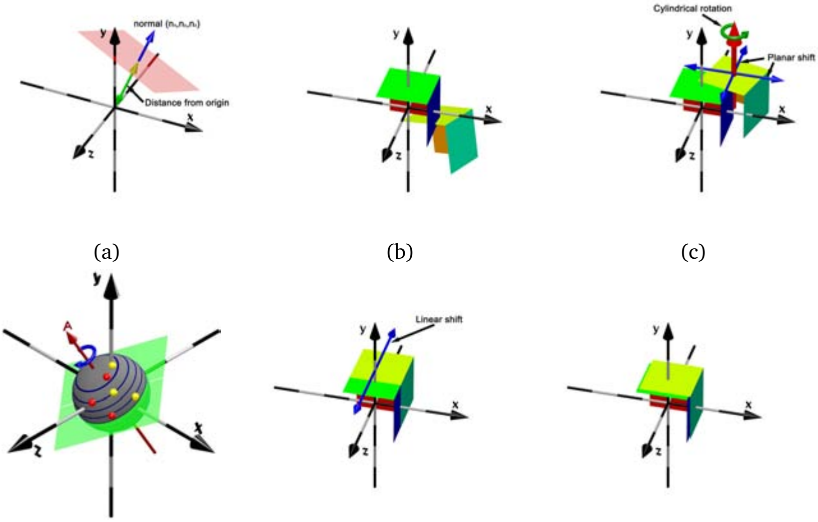
### 6.3.1 Planar matches

Planar objects are one of the most commonly found primitives both on industrial sites and buildings. A planar surface is represented by its normal $\mathbf{n}$ and the distance of the plane from the origin given by $\rho$. This representation is also called the Hesse form of the plane.

$$P = \begin{pmatrix} \mathbf{n} & \rho \end{pmatrix} \tag{6.1}$$

Although infinite planes do not exist in reality, it is convenient to work with them as they can be compactly represented without requiring the specification of boundary curves. As the final stage of Integrated adjustment uses original point-cloud, this implicitly brings in the information about the object boundaries.

For calculating the transformation between two coordinate frames we need at least 3 corresponding planes. To make the resulting combinatorial problem manageable we constrain each successive match by the geometric conditions imposed by the previous matches, a process detailed in the following paragraphs.



(a) (b) (c)

(d) (e) (f)

**Figure 6.1:** *Constrained correspondence search for planes (a) Parameters of a plane (b) before registration (c) after first match, a cylindrical rotation and a planar shift remains (d) search for planes which satisfy constraint of cylindrical rotation as seen on Gaussian sphere. The normals of the planes are represented as balls. (e) after second match, only a linear shift remains to be determined (f) after third match, all parameters are determined.*

### First match

For the first match of the planar objects we do not have any constraints. This means that all possibilities are equally valid and if we have $n_{fp}$ and $n_{rp}$ planes in the two scans, the number of possibilities is $n_{fp} \times n_{rp}$. The situation can be improved by using some heuristics that can prioritize matches based on criteria like number of points, error of fit etc. Additionally, if RANSAC is being used then any plane-pair at random can be picked.

If two planes $P_{f1}$ and $P_{r1}$ are found to be in correspondence with each other, then the rotation part of the transformation must align their normals i.e. $\mathbf{n}_{r1}$ with $\mathbf{n}_{f1}$. This can be accomplished by having a rotation of $\theta$ radians around axis $\mathbf{a}_r$, both of which are given by:

$$\theta = \cos^{\mathbf{-1}}(\mathbf{n_{r1}} \cdot \mathbf{n_{f1}}) \tag{6.2}$$

$$\mathbf{a} = \mathbf{n}_{r1} \times \mathbf{n}_{f1} \tag{6.3}$$

Similarly, the translation should take care of the differences in $\rho$ values.

$$\mathbf{t}_1 = (\rho_{f1} - \rho_{r1})\, \mathbf{n}_{f1} \tag{6.4}$$

Once this transformation $\mathbf{T}_1^P$ has been applied to the set of all planes we have:

$$\{P_r^1\} = \mathbf{T}_1^P \cdot \{P_r\} \tag{6.5}$$

### Second match

The second match for planes is constrained by the first match. There are two constraints (Figure 6.1(c)):

1. The rotation must be cylindrical around the axis given by $\mathbf{n}_{f1}$. This constraint ensures that the normals aligned during the first match (Section 6.3.1) remain undisturbed by the transformation of the second match.
2. The translation is constrained to only planar movements in the plane $P_{f1}$. There cannot be any movement along the normals which have been aligned by the first match, i.e. $n_{f1}$.

To illustrate the first constraint lets plot the normals of all planes on a Gaussian sphere. Each normal will map to a point which are shown as balls in Figure 6.1(d). The red balls represent the normal directions of reference planes i.e. $\{P_f\}$ and the yellow balls representing the planes in the scan being registered i.e. $\{P_r^1\}$. The blue arrow shows the direction of cylindrical rotation which is the same as $n_{f1}$ i.e the corresponding ball-pair aligned by the first match. Now if we cut the Gaussian sphere in planar slices along $n_{f1}$, then only the *yellow balls* which are in the same slice as some *red ball* can be mapped to each other by the cylindrical rotation. This narrows down the number of possible combinations drastically and speeds up the search process. Mathematically, this constraint can be given as:

$$\cos^{-1}\mathbf{n}_{f1}\cdot\mathbf{n}_{f2}\approx\cos^{-1}\mathbf{n}_{f1}\cdot\mathbf{n}_{r2} \tag{6.6}$$

If we look closely at the planar slices on the Gaussian sphere perpendicular to $\mathbf{n}_{f1}$ in Figure 6.1(d), we observe that as the planar slice moves away from the origin the circle resulting from planar slice (the blue rings in the figure) shrinks in diameter. This is a measure of the degeneracy of the match. The match on the great circle (the slice passing through origin) is orthogonal to the first match and will be geometrically the strongest. The situation worsens as the planar slice moves away from the origin, reaching the full degenerate case when the slice passes through the normals of the first match. This suggests a sub-clause to the first constraint, namely the angle between the plane-normals of the first match and this second one should be greater than zero i.e.

$$\cos^{-1}(\mathbf{n}_{f1}\cdot\mathbf{n}_{f2})>0 \tag{6.7}$$

This condition suggests that the matches from the slices closest to the origin (Figure 6.1(d)) should be tried first to get a geometrically well-determined solution.

The second constraint is not helpful in narrowing down the possible planar matches. This is because it can only affect translation, i.e. the value of $\rho$, but unless the rotation is fully determined, any value of $\rho$ can be attained through an appropriate translation in the plane $P_{f1}$.

At this stage if $P_{f2}$ is matched to $P_{r2}$, we have two correspondences for normal direction that fix the rotation. Translation is fully determined in the plane spanned by the normals of the corresponding planes, but there is still one degree of freedom left in the translation i.e. in the direction orthogonal to $\mathbf{n}_{f1}$ and $\mathbf{n}_{f2}$.

After the application of transformation matrix $\mathbf{T}_2^P$ we have the following situation:

$$\{P_r^2\} = \mathbf{T}_2^P \cdot \{P_r^1\} \tag{6.8}$$

### Third match

After the first two matches the rotation is fully determined. The third match is constrained by a linear motion in the direction of $\mathbf{l}_3$ which is perpendicular to both $\mathbf{n}_{f1}$ and $\mathbf{n}_{f2}$. i.e.

$$\mathbf{l}_3 \perp \mathbf{n}_{f1} \ \ \& \ \ \mathbf{l}_3 \perp \mathbf{n}_{f2} \ \Rightarrow \ \mathbf{l}_3 = \mathbf{n}_{f1} \times \mathbf{n}_{f2} \tag{6.9}$$

The above condition ensures that the $\rho$'s of the two plane-pairs aligned by the first two matches are not affected by the third match. Due to this condition, along with the fact that rotation is fixed, we have two conditions that must be satisfied for the third match:

1. As the rotation is fixed the normal directions of the plane-pair of this third match should be quite similar, meaning their dot product should be close to one ($t_s \approx 1$).
$$\mathbf{n}_{f3} \cdot \mathbf{n}_{r3} \geq t_s \tag{6.10}$$
2. As the translation is only along the direction $\mathbf{l}$, the normal direction of the third match should have a component along this direction i.e. their dot product should be greater than a given threshold ($t_p > 0$).
$$\mathbf{n}_{r3} \cdot \mathbf{l}_3 \geq t_p \tag{6.11}$$

The enforcement of these two constraints narrows down the search for possible matches, reducing the final computational cost. As rotation is already fixed, the only unknown is translation which is determined by the difference in $\rho$'s.

$$\mathbf{t}_3 = (\rho_{f3} - \rho_{r3})\,\mathbf{n}_{f3} \tag{6.12}$$

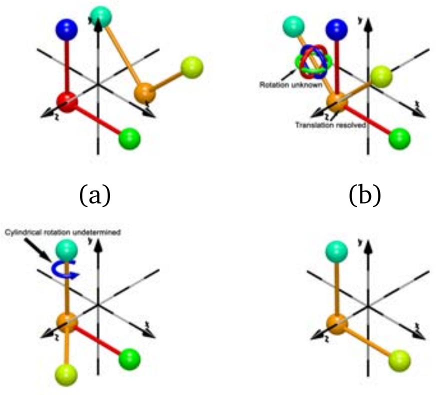
After applying this translation we get the final transformed set of planes.

$$\{P_r^3\} = \mathbf{T}_3^P \cdot \{P_r^2\} \tag{6.13}$$

After the third planar match the rigid body transformation is fully determined (Figure 6.1(f)) and we have an approximate registration solution.

## 6.3.2 Spherical matches

Spheres are often used as artificial targets, though they may occasionally be found as parts of industrial models. A sphere is given by its center $\mathbf{c}$ and radius $r$. Although $r$ does not contain any geometric information for registration (Chapter 5) it can used for filtering the matches based on the radius similarity constraint.



(a) (b) (c) (d)

**Figure 6.2:** *Constrained correspondence search for spheres (a) Before registration (b) after first match, translation is fixed but rotation is totally undetermined (c) after 2nd match, only a cylindrical rotation remains (d) after 3rd match, rigid transformation is fully determined*

The minimum number of sphere correspondences required to calculate registration is 3. Of course they must form a geometrically non-degenerate configuration. In the following sections we will formulate the constraints for each match.

### First match

For the first match we have only the constraint on the radius. i.e. $|r_{f1} - r_{r1}| < r_{th}$, where $r_{th}$ is a threshold close to zero. For the following sections we will refer to this constraint as the *radius similarity constraint* and we will use it for cylindrical and toroidal matches too.

If in the first match the sphere $S_{f1}$ matches the sphere $S_{r1}$, the translation is completely determined given by $\mathbf{t} = \mathbf{c}_{f1} - \mathbf{c}_{r1}$. Rotation is fully undetermined after the first match (Figure 6.2 (b)).

### Second match

For the second match we have two constraints:

1. Radius similarity constraint.
2. Distance similarity constraint. If the second match consists of spheres $S_{f2}$ and $S_{r2}$, then the distance of both from the center of $S_{f1}$ should be similar:

$$\|\mathbf{c}_{f2} - \mathbf{c}_{f1}\| \approx |\mathbf{c}_{r2} - \mathbf{c}_{r1}\| \tag{6.14}$$

As the translation was fixed by the first match, the second match gives us one corresponding vector direction, that can be used to calculate the rotation. The axis

$\mathbf{a}$ and angle $\theta$ of this rotation are given as follows:

$$\begin{aligned} \mathbf{a} &= (\mathbf{c}_{f2} - \mathbf{c}_{f1}) \times (\mathbf{c}_{r2} - \mathbf{c}_{r1}) && (6.15) \\ \theta &= \cos^{-1}\left((\mathbf{c}_{f2} - \mathbf{c}_{f1}) \cdot (\mathbf{c}_{r2} - \mathbf{c}_{r1})\right) && (6.16) \end{aligned}$$

After the application of this transformation we have the situation as shown in Figure 6.2 (c).

#### Third match

After the first two matches the translation is fully determined while a cylindrical rotation still remains to be determined. The third spherical match is constrained by the following constraints:

1. Radius similarity constraint (See 6.3.2).
2. Cylindrical rotation around the axis $(\mathbf{c}_{f2} - \mathbf{c}_{f1})$.
3. Distance similarity constraint as defined before for the second match. ($\|\mathbf{c}_{f3} - \mathbf{c}_{f1}\| \approx |\mathbf{c}_{r3} - \mathbf{c}_{r1}\|$)

The second constraint can be enforced using the Gaussian sphere based technique described above (Section 6.3.1). The third match gives us a fully resolved solution of the registration problem (Figure 6.2 (d)).

### 6.3.3 Cylindrical matches

A cylinder in our system in represented by 7 parameters, 3 for the axis direction $\mathbf{a}_c$, three for the point closest to the origin $\mathbf{p}_c$, and one for the radius of the cylinder, $r_c$ (Figure 6.3(a)).

$$C = \begin{pmatrix} \mathbf{a}_c & \mathbf{p}_c & r_c \end{pmatrix} \tag{6.17}$$

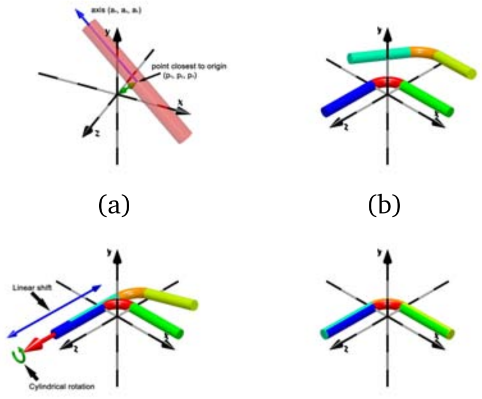
For a cylinder the degrees of freedom are 5, therefore we have two conditions on this over-parameterized representation

$$\|\mathbf{a}_c\| = 1 \qquad \& \qquad \mathbf{a}_c \cdot \mathbf{p}_c = 0 \tag{6.18}$$

The first condition requires the magnitude of the axis to be unity while the second condition ensures that $\mathbf{p}_c$is a point on cylinder closet to origin. The minimum number of cylindrical matches required for determination of all registration parameters is 2.

#### First match

For the first match we have only radius similarity condition for cylinders i.e. $|r_{f1} - r_{r1}| \leq t_r$ (Figure 6.3 (b) ).



(a) (b) (c) (d)

**Figure 6.3:** *Constrained correspondence search for cylinder (a) Parameters of the cylinder, we use 7 parameters with two constraints (b) before registration (c) after first match, a cylindrical rotation and linear shift remains (d) after second match, all parameters are determined.*

Once we have one pair of matching cylinders $C_{f1}$ and $C_{r1}$, we can determine the required transformation up to a cylindrical rotation and a linear shift. The rotation of this transformation is around the axis of $\mathbf{a}_{f1} \times \mathbf{a}_{r1}$, with an angle of $\cos^{-\mathbf{1}} \mathbf{a}_{f1} \cdot \mathbf{a}_{r1}$. The shift is given by

$$\mathbf{t}_1 = (\mathbf{p}_{f1} - \mathbf{p}_{r1}) - ((\mathbf{p}_{f1} - \mathbf{p}_{r1}) \cdot \mathbf{a}_{f1})\, \mathbf{a}_{f1} \tag{6.19}$$

Once the transformation obtained by combining this rotation and translation has been applied to the cylinder, we have:

$$\{C_r^1\} = T_1^C \cdot \{C_r^0\} \tag{6.20}$$

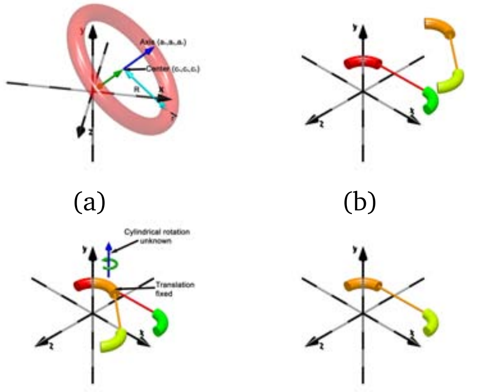
### Second match

The second match has to satisfy three constraints (Figure 6.3 (c) ):

1. The radius difference should be smaller than a certain threshold i.e. $|r_{f2} - r_{c2}| \leq t_r$.
2. The rotation between $C_{r2}{}^1$ and $C_{f2}$ should be a cylindrical rotation around $\mathbf{a}_{f1}$. For enforcing this constraint we can use the same technique as Section 6.3.1.
3. Once we have rotated $C_r^1$ to $C_f^1$ there is still a translation $\mathbf{t}_2$ that needs to be determined. It is constrained to be a linear shift along $\mathbf{a}_{f1}$. i.e. $\mathbf{t}_2 = \lambda \mathbf{a}_{f1}$ where $\lambda$ is a parameter to be determined. For deriving an expression for $\lambda$ we proceed as follows.

$$\mathbf{p}_{f2} = \mathbf{p}_{r2} = \mathbf{p}_{r1} + \lambda \mathbf{a}_{f1} - (\mathbf{a}_{r2} \cdot (\mathbf{p}_{r1} + \lambda \mathbf{a}_{f1}))\, \mathbf{a}_{r2} \tag{6.21}$$

$$\mathbf{p}_{r1} \perp \mathbf{a}_{r2} \Rightarrow \mathbf{a}_{r2} \cdot \mathbf{p}_{r1} = 0 \tag{6.22}$$



(a) (b) (c) (d)

**Figure 6.4:** *Constrained correspondence search for torus (a) Parameters of the torus, we use 8 parameters with one constraint (b) before registration (c) after first match, only a cylindrical rotation remains (d) after 2nd match, all parameters are determined.*

$$\lambda = \frac{\mathbf{p}_{f2} - \mathbf{p}_{r1}}{\mathbf{a}_{f1} - (\mathbf{a}_{f1} \cdot \mathbf{a}_{f2})\,\mathbf{a}_{f2}} \tag{6.23}$$

From the above equation we will get three different values for $\lambda$, and if they are approximately equal $\lambda_x \approx \lambda_y \approx \lambda_z$ we have a unique solution and its possible to align the cylinders while satisfying the constraints. Otherwise we will discard this match and try another one.

After the 2nd match we have two axial directions which fully determine the rotation and two positions on these axes can be used to calculate the translation. By combining the rotation with the translation we get the full transformation $\mathbf{T}_2^C$.

Infinite cylinders can be treated as infinite lines in 3D with a finite radius. For transforming such lines Plücker coordinates with dual-quaternions have been extensively used. It is possible to use that representation for our problem too. Although we do not see any special advantage in that, expect may be compactness of the derived results as the rotation and the translation can be treated together in Plücker coordinates using dual quaternion (Daniilidis and Bayro-Corrochano, 1996).

### 6.3.4 Toroidal matches

Each torus $T$ is represented by 8 parameters, 3 for normal direction of the plane containing the torus $\mathbf{a}_t$, the central point $\mathbf{p}_t$, minor radius $r_t$and major radius $R_t$ (Figure 6.4(a)).

$$T = \begin{pmatrix} \mathbf{a}_t & \mathbf{p}_t & R_t & r_t \end{pmatrix} \tag{6.24}$$

There is only one constraint namely $\|\mathbf{a}_t\| = 1$.

### First match

For the first match we have only two constraints:

1. $|r_{f1} - r_{r1}| \leq t_r$ where $t_r$ is a threshold for minor-radius similarity
2. $|R_{f1} - R_{r1}| \leq t_R$ where $t_R$ is a threshold for major-radius similarity.

Once we have a match between a pair of tori, the rotation is determined up to a cylinderical ambiguity around the axis $\mathbf{a}_{f1}$ while the translation is fully determined. For this purpose the torus may be considered as a combination of a sphere placed at $\mathbf{p}_t$, and a cylinder in the direction of $\mathbf{a}_t$.

### Second match

For the second match we are looking for information to fix a cylindrical rotation with the axis of rotation being equal to $\mathbf{a}_{f1}$. To do the search we use the Gaussian sphere based method that was used for finding planes in Section 6.3.1.

## 6.3.5 Mixed objects.

In real scenarios we cannot expect that only one type of objects will be present in a given scene. It would always be a combination of objects. In such scenarios it is necessary to use the most constraining matches first, so that we can come to a solution without doing a lengthy search. Here are some rules of thumb for guiding the search in the case of mixed objects.

- Sort objects according to their constraint strength, this will result in the following order: torus, cylinder, plane and sphere.
- Within each object class sort the objects based on some confidence measure. For example number of points used to fit an object, the average residual for least square fitting, or the quality of the parameter estimation coming from the stage of model fitting (Chapter 4).

# 6.4 Comparison

We implemented RANSAC based correspondence search. Two versions of the methods were implemented: one using standard RANSAC and the other incorporating the presented set of constraints. A detailed algorithm for incorporating constraints in the search for planar corresponding objects is given in Algorithm 2.

**Algorithm 2** RANSAC based correspondence search for planes

```
Inputs: Point clouds = $\mathbf{\Theta}_r$ and $\mathbf{\Theta}_f$ point normals = $\mathbf{\Pi}_r$ and $\mathbf{\Pi}_f$ modeled planes = $\{P\}_r$ and $\{P\}_f$ number of maximum iterations = $iter_{max}$ %age overlap = $\eta_{rf}$ distance threshold = $d_{th}$ angle threshold = $\theta_{th}$
Initialize counter $c = 0$
while $c < iter_{max}$ do
    First match
    Pick first plane-pair at random. $P_{f1} = \{P\}_f[rand]$ and $P_{r1} = \{P\}_r^0[rand]$.
    Apply the transformation of the first pair to all planes being registered $\{P\}_r^1 = \mathbf{T}_1 \cdot \{P\}_r^0$
    Second match
    Pick a fixed plane at random $P_{f2} = \{P\}_f[rand]$ so that its normal has some angle with the first match $\cos^{-1}(\mathbf{n}_{P_{f1}} \cdot \mathbf{n}_{P_{f2}}) > 0$
    The plane from the registered set $P_{r2}$ must satisfy the cylindrical rotation constraint $\cos^{-1}(\mathbf{n}_{P_{f1}} \cdot \mathbf{n}_{P_{f2}}) = \cos^{-1}(\mathbf{n}_{P_{f1}} \cdot \mathbf{n}_{P_{r2}})$
    Apply the transform from the 2nd match to all planes in the registered set $\{P\}_r^2 = \mathbf{T}_2 \cdot \{P\}_r^1$
    Third match
    Pick fixed plane at random $P_{f3} = \{P\}_f[rand]$ so that it has a component perpendicular to the plane spanned by first two matches $\cos^{-1}((\mathbf{n}_{P_{f1}} \times \mathbf{n}_{P_{f2}}) \cdot \mathbf{n}_{P_{f3}}) > 0$ and
    Pick the plane from registered set $P_{r3}$ so that it has a normal in the same direction as $P_{f3}$ i.e. $\cos^{-1}((\mathbf{n}_{P_{f3}} \cdot \mathbf{n}_{P_{r3}}) \approx 0$
    Apply the transform from the 3rd match to all planes in the registered set. $\{P\}_r^3 = \mathbf{T}_3 \cdot \{P\}_r^2$
    Verify
    Transform $\mathbf{\Theta}_r$ using the calculated values.
    Calculate the overlap between $\mathbf{\Theta}_r$ and $\mathbf{\Theta}_f$ by counting the number of points which have distances less than $d_{th}$ and whose normals from $\mathbf{\Pi}_r$ and $\mathbf{\Pi}_f$ have angles less than $\theta_{th}$.
    if Calculated overlap $>= \eta_{rf}$ then
        Return the calculated transform $\mathbf{T}$
    else
        Not a valid match. Continue with the next iteration.
    end if
end while
Return Failure. Number of specified iterations exceeded before we could find a transform with specified overlap.
```

To compare the theoretical improvements we generated synthetic data sets with different number of planes and cylinders. Each dataset was processed using both search strategies. As RANSAC is a random sampling procedure each search was repeated 50 times to get an average number of iterations required to find the solution. The results are presented in Table 6.1. There we see for cylinders an improvement of an order of 50. While for planes it varies between a factor of 100 to 1000. Of course this improvement is highly dependent on the site and type and number of geometric primitives found there, but on average we can expect significant improvements in search performance.

**Table 6.1:** *Comparison of the number of iterations required for different methods of correspondence search*

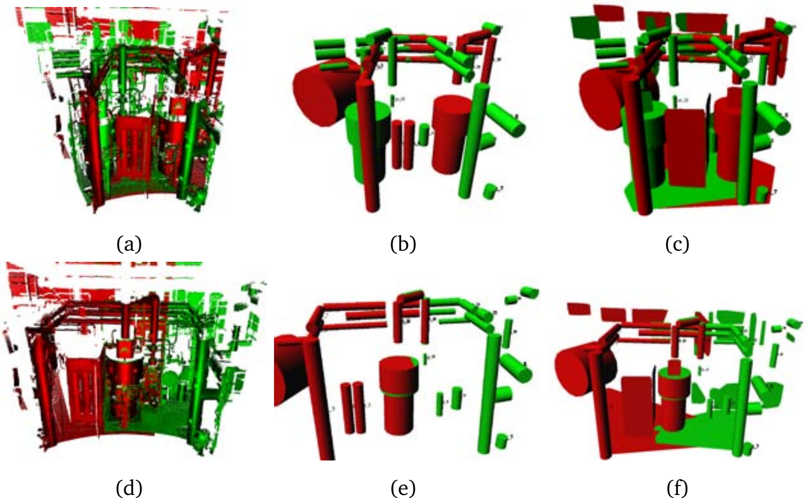
| **Method** | **Number of objects** | | | | | |
|---|---|---|---|---|---|---|
| | 5 | 10 | 30 | 50 | 70 | 100 |
| **For Cylinders** | | | | | | |
| Brute force search | 400 | 8100 | $7\times10^5$ | $6.\times10^6$ | $2\times10^7$ | $9.8\times10^7$ |
| RANSAC without constraints | 28.7 | 138.9 | 1062 | 1887 | 3228 | 5028 |
| RANSAC with constraints | 1.12 | 3.68 | 16.5 | 21.3 | 48.8 | 75 |
| RANSAC $\frac{without}{with}$ | 25.6 | 37.7 | 64.3 | 88.6 | 66.0 | 67.04 |
| **For Planes** | | | | | | |
| Brute force search | 3600 | $5\times10^5$ | $6\times10^8$ | $10^{10}$ | $10^{11}$ | $10^{12}$ |
| RANSAC without constraints | 164 | 1463 | $5\times10^4$ | $8.6\times10^4$ | $9.1\times10^4$ | $> 10^5$ |
| RANSAC with constraints | 4.6 | 7.5 | 21.2 | 67.4 | 92.7 | 122 |
| RANSAC $\frac{without}{with}$ | 35.6 | 195 | 2358 | 1275 | 981 | 819 |

## 6.5 Results

We applied the presented method to some industrial data sets. In Figure 6.5 we see the situation before and after the first step of automatic approximate registration. For this scenario we used only cylinder based automatic search. We had $17$ cylinders in the first scan and $26$ in the second scan. As only two matches are required for an approximate alignment, a brute force search would have required $17 \times 26 \times 16 \times 25 = 176800$ matches. In contrast our search algorithm was able to find the solution in $20$ matches. This reduction was possible by exploiting the geometric constraints and by breaking the search once a *good enough* solution was found. The transformation parameters from this procedure are only approximate. This is clear from Figure 6.5, and is emphasized in Figure 6.6, where a zoom on the point-clouds after the step of approximate alignment shows remaining errors. In Figure 6.6(b) we see a significant improvement after the Integrated adjustment.

## 6.6 Conclusions

We have presented a method for automatic registration of point clouds using an efficient search for corresponding objects. Although this methodology is not


(a) (b) (c)

(d) (e) (f)

**Figure 6.5:** *Results of processing an industrial data-set (a), (b), (c), point-clouds, cylinders, and cylinders+planes before automatic approximate registration. (d), (e), (f) after automatic approximate registration. In all figures* $\mathbf{\Theta}_f$ *is in red and* $\mathbf{\Theta}_r$ *is in yellow.*

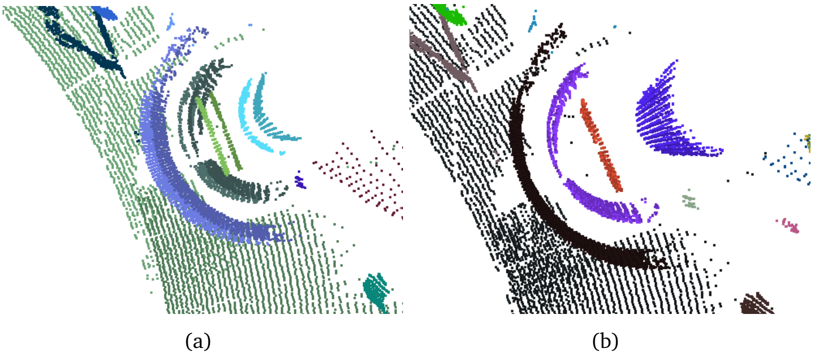

(a) (b)

**Figure 6.6:** *The improvements obtained by integrated adjustment (a) after approximate alignment through constrained correspondence search (b) after least square Integrated adjustment*

general and cannot be applied to all scenes, it can reduce the manual labor for registration of industrial point clouds. This procedure can be extended to the registration of images with point-clouds of industrial scenes by finding correspondences between the back-projected model contours and image edges.

Chapter 7

# Constrained CSG fitting to point clouds & images

> A picture is worth a thousand words.
>
> Napoleon Bonaparte

> The ability to simplify means to eliminate the unnecessary so that the necessary may speak.
>
> Hans Hofmann

In this chapter we present a method for the 3D reconstruction of industrial sites using a combination of images and point clouds with a motivation of achieving higher levels of estimation accuracy and model completeness. As discussed in the previous chapters dense point clouds for 3D reconstruction can be easily acquired using a laser scanner. Compared to images the point clouds contain explicit 3D information, as a result they have a much higher potential for the automation of 3d reconstruction. This was demonstrated in Chapter 3 for automatic plane and cylinder detection through the Hough transform. However, due to the measurement principle employed by the laser scanners and their limited point density, the information on the sharp edges is not very reliable. As images have better information content on the edges of the objects, it makes them a complementary source of data. In addition images are required for visual interpretation, texture mapping, and modeling parts not visible in the point clouds. Furthermore, the acquisition of images is more flexible, and the cost and time required for it is much lower than that of laser scanning, making their combined use essential for a cost-effective solution. As many new laser scanners are coming with integrated cameras, it provides another motivation for using images for modeling. These reasons led us to develop a modeling strategy that uses both images and point clouds in combination with a library of CAD primitives commonly found in industrial scenes. The primitives in this library are represented as CSG (Constructive Solid Geometry) models.

For the fitting of CSG objects to the images we expect their exterior orientations to be known. The CSG objects are fitted in a least squares adjustment that minimizes the distance between the projected contours of the model to the measurements on the visible edges in the images. These measurements increase the estimation quality of the modeling, especially for those portions where the information in the point cloud is either noisy or is missing. This will be shown in the two fitting experiments performed on relatively simple scenarios.

The approach of explicit specification of constraints, that was presented in Chapter 4, had some serious limitations. It could only handle linear constraints. Moreover, there was no easy way to specify external constraints (linking the parameters of one CSG tree to those of another). To overcome these limitations in this chapter we present a fitting method that uses constraints through Lagrange multipliers. This enables us to handle both nonlinear equality and inequality constraints. Additionally, the specification of external constraints can be easily handled. We present the mathematical formulation of some of the commonly encountered constraints and give the details of their implementation.

The results of applying this method to data-sets from industrial sites are presented showing the complementary nature of the point cloud and the image data. An analysis of the improvements in the quality of the 3D reconstruction shows the benefits of the presented approach.

## 7.1 Introduction

We discussed the advantages of using point clouds for modeling in Chapter 1. Although, the advent and rapid developments in the field of high speed 3D data capture might give an impression that the images are now outdated, in reality images still provide a complementary source of information. Here we outline the reasons that motivated us to make a combined use of images and point clouds.

**Accurate edge information** Laser scanners and cameras employ two different physical measurement principles which lead to different strengths and weaknesses. A camera focuses the backscattered light using a lens on a photosensitive sensor (CCD or CMOS). In this arrangement the resolution of the system is limited by the distance of the object from the camera, the number of pixels on the sensor, and the Modulation Transfer Function (MTF) of the lens. In contrast, the laser scanners are based on a point measurement device, which emits a laser pulse with a finite beam-width. The time of flight or phase-shift of the reflected beam is measured and converted to distance. This point measurement “head” is scanned in two directions using mechanical devices to achieve an area-scan instrument. As pointed out by Lichti (2004), in addition to the internal measurement errors, the beam-width of the emitted laser limits the spatial resolution of the laser scanners. The typical beam-widths of terrestrial laser scanners varies from a few mm to tens of mm. This is not a serious problem for the measurement of smooth surfaces (e.g. insides of a cylindrical surface) but it results in a serious degradation of the measurements on those areas of the surface which have rapidly varying details like the edges of the objects. This is a direct result of a bigger beam-width and can be compared to a blurred image from an out of focus camera. Furthermore, the quality of the range measurement obtained from the laser scanners is affected by the reflectivity, the color and the roughness of the surface being measured (Böehler et al., 2003; Clark and Robson, 2004). As images have a better information content on the edges of the objects, they provide a complementary data source and their inclusion in model fitting should give better results.

**Easy and fast capture** Although the laser scanners are improving in speed with each new generation of instruments, they will remain for foreseeable future far behind the cheap, fast and flexible data capture of the cameras. The combined use of images and point clouds provides an economical solution for complete as built modeling. The fact that most new laser scanners are coming with integrated cameras provides another opportunity to use this readily available data for modeling.

**Visualization** Most applications of as-built modeling require or can benefit from a realistic rendering of the 3D world, which requires registered imagery for texture mapping. Additionally, compared to the raw point clouds colored images are more intuitive for human interpretation and can be used for this purpose during manual or semiautomatic processing.

**Validation of modeling** Images provide a way to check the validity and completeness of the final model. Superimposition of the back-projected contours of the modeled surfaces on the images provide a convenient tool for this purpose. For a user it can be easier to identify the un-modeled surfaces in the images compared to doing the same job with the point clouds.

**Improved estimation** As will be shown in Section 7.5, the inclusion of images can provide additional information which is not available from the point clouds and this can result in a better quality of estimation during model fitting. This is especially true for the bounded objects, because the parameters determining the bounds are estimated better when the edges belonging to the faces not covered by the point cloud are included in the estimation procedure. For example, the sizes of a box or the length of a cylinder are poorly estimated from the points cloud, as in most cases not all the faces are fully captured. In such cases the situation improves significantly if the edge measurements from the images are used in combination with the surface measurements from the point cloud.

Based on the above observations of the complementary nature of the images and the point clouds we decided to use both data sources simultaneously.

## 7.2 Model Fitting

In Chapter 4 we presented an approach for the fitting of CSG models to the point clouds with explicit constraints. This enabled us to use standard unconstrained least squares optimization methods for solving a constrained problem. But, as we noted there, this had some serious limitations. Only linear constraints could be included. The inclusion of nonlinear constraints like the one on the length of a quaternion or axis of the cylinder being unity were handled outside the least squares method. Similarly, inequality constraints were limited to only simple bounds. Moreover, specification of external constraints was not possible.

Based on all of the above limitations we decided to move to the constrained optimization methods. This enables us to specify both linear and non-linear constraints. The possibility to include inequalities provides a way to specify the feasible parameter domains in a more flexible way. For example, the angle between two cylinders can be constrained to be more than 85 and less than 95 degrees. External constraints, which link the parameters of one CSG tree to those of another, can be directly specified. Overall, the use of constrained optimization provides a better and more flexible approach.

### 7.2.1 Observation equations with constraints

In this section, we will discuss the constrained least squares method for linear equations and linear equality constraints. The extension of this method to the non-linear models is based on the linearization of the involved equations through Taylor series expansion.

Consider a mathematical model with $n$ observations, $p$ unknown parameters, and $m$ constraints. The observation equations are given by:

$$\begin{array}{cccc} \mathbf{A} & \mathbf{x} & = & \mathbf{b} \\ (n \times p) & (p \times 1) & & (n \times 1) \end{array} \tag{7.1}$$

where $\mathbf{A}$ is the design matrix, $\mathbf{x}$ are the unknown parameters being estimated, and $\mathbf{b}$ are the values of the observations. If a weight matrix $\mathbf{W}$ is given the above system changes to $\mathbf{WAx} = \mathbf{Wb}$.

The constraints which express the relationships between the different elements of $\mathbf{x}$ are given by:

$$\begin{array}{cccc} \mathbf{C} & \mathbf{x} & = & \mathbf{d} \\ (m \times p) & (p \times 1) & & (m \times 1) \end{array} \tag{7.2}$$

where $\mathbf{C}$ is the design matrix for the constraints and $\mathbf{d}$ are their values.

The above system can be solved by introducing a set of $m$ new variables $\lambda_i$ called Lagrange multipliers. Their inclusion converts the above system to the following:

$$(\mathbf{WAx} - \mathbf{Wb}) + \mathbf{\Lambda}^T(\mathbf{Cx} - \mathbf{d}) = 0 \tag{7.3}$$

where $\mathbf{\Lambda}^T$ is a diagonal matrix whose diagonal entries are given by $m$ Lagrange multiplier.

The estimated solution $\widehat{\mathbf{x}}$, obtained by solving Equation 7.3, is given by the following equations (Wong, 1980):

$$\widehat{\mathbf{x}} = \widehat{\mathbf{x}}_o - \widehat{\mathbf{x}}_c \tag{7.4}$$

$$\widehat{\mathbf{x}}_o = \left(\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}\mathbf{W}\mathbf{b} \tag{7.5}$$

$$\widehat{\mathbf{x}}_c = \left(\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\left(\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{C}^{\mathrm{T}}\right)^{-1}\left(\mathbf{C}\widehat{\mathbf{x}}_o - \mathbf{d}\right) \tag{7.6}$$

The above equations show that the solution of a constrained system of equations consists of two parts: the first part $\widehat{\mathbf{x}}_o$ is determined by the observations. The second part $\widehat{\mathbf{x}}_c$ just introduces a shift to $\widehat{\mathbf{x}}_o$, so that the final solution obeys all the constraints.

A similar behavior is shown by the covariance matrix of the estimated parameters $\mathbf{\Sigma}_{\widehat{\mathbf{x}}}$ which is given by:

$$\mathbf{\Sigma}_{\widehat{\mathbf{x}}} = \mathbf{\Sigma}_{\widehat{\mathbf{x}}_o}\left(\mathbf{I} - \mathbf{\Sigma}_{\widehat{\mathbf{x}}_c}\right) \tag{7.7}$$

$$\mathbf{\Sigma}_{\widehat{\mathbf{x}}_o} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\right)^{-1} \tag{7.8}$$

$$\mathbf{\Sigma}_{\widehat{\mathbf{x}}_c} = \mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\left(\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{C}^{\mathrm{T}}\right)^{-1}\mathbf{C}\left(\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\right)^{-1} \tag{7.9}$$

From the above equations we see that the constraints add a multiplicative term $(\mathbf{I} - \mathbf{\Sigma}_{\widehat{\mathbf{x}}_c})$ to the covariance matrix $\mathbf{\Sigma}_{\widehat{\mathbf{x}}_o}$ that would have resulted if the constraints were ignored. For a constrained system the variance of those parameters decreases whose estimation has improved because of the additional information provided by the constraints.

For solving non-linear observation equations with non-linear constraints, both need to be linearized and then solved by the machinery outlined above. This is an iterative procedure and replaces $\mathbf{A}$ and $\mathbf{C}$ by the Jacobian matrices of the observations and the constraints ($\mathbf{J}_A$ and $\mathbf{J}_C$) respectively. The direct linearization does not have good convergence properties. Different methods have been proposed in the literature of constrained nonlinear optimization (some times called nonlinear programming) to get better global convergence properties (Bertsekas, 1995; Dennis and Schnabel, 1996; Luenberger, 1984). We have decided to use a Gauss-Newton based method for solving constrained nonlinear optimization problem, which uses an active set strategy for the inequality constraints. For more details see Eriksson and Wedin (2004); Lindström (1983, 1984); Lindström and Wedin (1986).

Due to the active-set strategy the inequality constraints are also treated as equality constraints when they are active, otherwise they are ignored. Once the final solution is reached the covariance matrix is estimated using Equation 7.7, but with the Jacobian matrices $\mathbf{J}_A$ and $\mathbf{J}_C$ replacing $\mathbf{A}$ and $\mathbf{C}$ respectively.

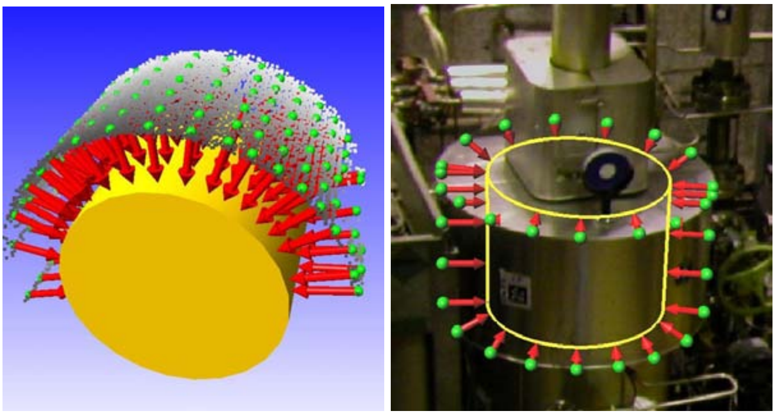

(a) (b)

**Figure 7.1:** *(a) Fitting of a model to the point cloud, the sum of squares of the orthogonal distances of the points to the estimated model surface is minimized (b) Fitting to the images minimizes the sum of squares of the distances of the measured points (the green spheres) from the back-projected contours (shown as yellow outline) of the model being fitted.*

### 7.2.2 Fitting of a CSG model

For fitting of a CSG model to a given point cloud, the problem formulation is quite similar to the one given in Chapter 4. We minimize the sum of the squares of the orthogonal distances of the points from the surface of the model being fitted, while enforcing the equality and the inequality constraints between the parameters of the CSG model:

$$\min \sum_{i=1}^{N} \Omega^2 \left[p_i, \Gamma\left(\tau_1, \tau_2, \ldots, \tau_m\right)\right] \qquad i = 1, 2, \ldots, n \tag{7.10}$$

$$g_j(\tau_1, \tau_2, \ldots, \tau_M) = 0 \qquad j = 1, 2, \ldots, r \tag{7.11}$$

$$h_k(\tau_1, \tau_2, \ldots, \tau_M) > 0 \qquad k = 1, 2, \ldots, s \tag{7.12}$$

$\Omega$ defines the shortest distance of a given point $p_i$ from the surface of the CSG model $\Gamma$ if the point comes from a 3D point cloud or the distance to the closest back projected edge if the point is measured in an image (Figure 7.1). The CSG object has $m$ shape and pose parameters given by $\tau_1, \tau_2, \ldots, \tau_m$. There are $r$ equality constraints given by $g_j$'s and $s$ inequlity constraints given by $h_k$'s. These constraints are enforced using the Lagrange multiplier based method discussed in the last section.

For the calculation of $\Omega$ (the orthogonal distance of a point from CSG object's surface), we have already presented a comparison of different numerical methods in Chapter 4.

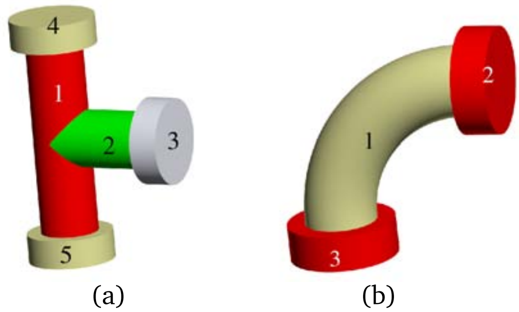


**Figure 7.2:** *Examples of the geometric constraints for CSG objects (a) Flanged T junction (b) Flanged curve*

For fitting a CSG model to the images we follow the technique presented by Ermes et al. (1999); Tangelder et al. (2003). There is one exception in our approach, because we do not know a priori the correspondences between the image measurements and the back projected contours of the CSG model. Due to this missing information we follow an iterative procedure, where before each new iteration the measurements are assigned to the closest back-projected contour.

Each point measurement in an image gives us a ray in 3D. Given a set of images with measured points we want to estimate those values for the CSG parameters that result in the minimum distance (in a least squares sense) between all these rays and the estimated CSG model. Alternatively, the ray to the body distance can be calculated in the image space. There the distance in pixels between an image measurement and the closest back-projected contour of the CSG model is calculated. The back projection must have a mechanism for hidden-line removal, so that the effects of self and external occlusions are taken into account. We use ACIS (Spatial, 2004) to compute the hidden line projection of the model in the images. For an example see Figure 7.1(b).

## 7.3 Types of constraints

A constraint is a geometric or topological relationship that limits the permissible values for the parameters and other properties of the objects in a design. It is a powerful way to encode a priori information and to specify the design intent and limitations both during the designing of a new model as well as during the reconstruction/modeling of an existing object. As each constraint decreases the degrees of freedom, it narrows down the search space by shrinking the feasible set of parameter values. As a result it can lead to better estimation and convergence properties during fitting.

In the following paragraphs we will present some of the commonly encountered constraints for the CSG objects found in the industrial environments.

### 7.3.1 Parameter constraint

This constraint can be used to specify the relationship between two parameters, either of the same primitive or of two different primitives. If these primitives comprise sub-parts of a single CSG tree then the resulting constraint is an *Internal constraint*; otherwise if both the primitives belong to two different CSG trees then the constraint is an *External constraint*.

The general form of the parameter constraint can be given by:

$$p_1 = c_2 p_2 + d \tag{7.13}$$

Where $p_1$ and $p_2$ are the parameters, $c_2$ is a constant scaling factor and $d$ is a constant offset. Some simple variations of this constraint are:

1. $p_1 = p_2$ (both parameters have the same value)
2. $p_1 = d$ (parameter 1 is given a fixed value)
3. By including inequalities this constraint can be used to specify bounds. For example $d_l < p_1 < d_h$.

For example, for a flanged T-junction the radii of the cylinders 1 & 2 are related by this constraint (Figure 7.2(a)). Similarly, by using an inequality parameter constraint it can be enforced that the radius of the external cylinders (3, 4, & 5) is always greater than the radius of the internal cylinders (1 & 2).

### 7.3.2 Rotation constraint

This constraint is used to specify the angle between two coordinate axes $\mathbf{a}_1$ and $\mathbf{a}_2$, where both of them can be one of the three axes of the local coordinate systems of the primitives 1 and 2.

Mathematically this constraint can be expressed as follows:

$$\cos^{-1}(\mathbf{a}_1.\mathbf{a}_2) + c_1 p_1 = c_2 p_2 + \theta \tag{7.14}$$

The simplest form just gives the angle (in radians) between the two axes:

$$\cos^{-1}(\mathbf{a}_1.\mathbf{a}_2) = \theta \tag{7.15}$$

The above equation says that $\mathbf{a}_1$ can rotate around $\mathbf{a}_2$ in a cylindrical way, keeping a fixed angle of $\theta$. By specifying two such constraints, the coordinate system of both entities are fixed in rotation relative to each other, which can be used for example for fixing the orientation of a box with respect to another box.

By including inequalities this constraint can be extended for specifying angle bounds:

$$\theta_l < \cos^{-1}(\mathbf{a}_1.\mathbf{a}_2) < \theta_h \tag{7.16}$$

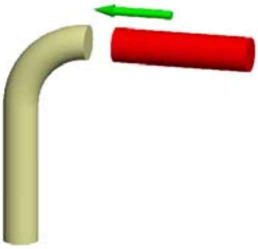


**Figure 7.3:** *Connection constraint for CSG fitting. The red pipe is to be connected with the yellow curve.*

This constraint always results in one equation or two inequalities (one each for upper and lower bounds). For example the orthogonality of the cylinders 1 and 2 for a flanged T junction can be specified using this constraint (Figure 7.2(a)).

### 7.3.3 Translation constraint

The translation constraint gives the relationship between the translations of two primitives. Mathematically it is given as follows:

$$\mathbf{t}_1 + c_1\mathbf{a}_1 = \mathbf{t}_2 + c_2\mathbf{a}_2 \tag{7.17}$$

Where $\mathbf{t}_1$ and $\mathbf{t}_2$ are the translations of the primitives and $\mathbf{a}_1$ and $\mathbf{a}_2$ are the directions for translation.

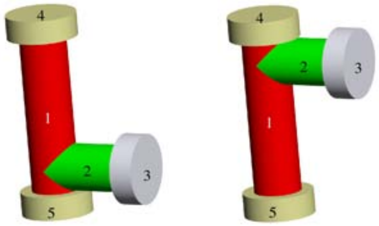
We can have the following three cases with different number of effective degrees of freedom (DOF):

- $\mathbf{t}_1 = \mathbf{t}_2$ (DOF = 0, active constraints = 3)
- $\mathbf{t}_1 = \mathbf{t}_2 + c_1\mathbf{a}_1$ (primitive 1 can move along $\mathbf{a}_1$, DOF = 1, active constraints = 2)
- $\mathbf{t}_1 = \mathbf{t}_2 + c_1\mathbf{a}_1 + c_2\mathbf{a}_2$ (primitive 1 can move in the plane spanned by $\mathbf{a}_1$ and $\mathbf{a}_2$, DOF = 2, active constraints = 1)

When one of the primitives is a torus, then additionally an angle $\alpha$ can be specified, which ties the position of the object to lie along one of the ends of the torus. In that case the equation becomes:

$$\mathbf{t}_1 = \mathbf{t}_2 + \mathbf{R}_t \begin{pmatrix} r_1 \cos\alpha \\ r_1 \sin\alpha \\ 0 \end{pmatrix} \tag{7.18}$$

Where $\mathbf{R}_t$ is the rotation matrix for the torus and $r_1$ is its major radius. This variation of the translation constraint is used to place cylinders 1 and 2 at the two ends of a curve in Figure 7.2(b).



(a) (b)

**Figure 7.4:** *Bounded translation constraint (a) The lower limit for the translation of cylinder 2 (b) The upper limit of the translation*

### 7.3.4 Connection constraint

A connection constraint is a combination of the following sub-constraints to specify the connection of two cylinders or the connection of one cylinder and one torus:

1. A parameter constraint (radii should be equal).
2. A translation constraint which can be one of the following:
   - If both the primitives being connected are cylinders, translate along the axis of the first cylinder equal to its length to find the position of the second cylinder).
   - If one of the primitives is a torus, translate along the curve of the torus equal to its angle to find the position of the cylinder (Figure 7.3).
3. A direction constraint (both axial directions should be the same, as given by the axial direction of the cylinder or the tangential direction of the toroidal curve).

An example of a situation where the connection constraint can be useful is shown in Figure 7.3. The red cylinder is to be connected to the yellow curve, and this relationship can be enforced by using a connection constraint.

### 7.3.5 Bounded translation constraint

This is an inequality constraint which is an extension of the translation constraint. The position of one primitive is constrained to lie between two limits which depend on the parameters of another primitive. For example in a flanged T-junction the middle cylinder (number 2 in Figure 7.2(a) ) is constrained to lie between the limits defined by the positions of the top two cylinders (4 & 5) . The extreme positions possible are shown in Figure 7.4. The middle cylinder should not go out of the bounds set by these two extreme positions. This constraint can be enforced by using the following inequalities:

$$(\mathbf{t}_1 + r_2\mathbf{a}_1) < \mathbf{t}_2 < (\mathbf{t}_1 + (l_1 - r_2)\,\mathbf{a}_1) \tag{7.19}$$

Where $l_1$ is the length of cylinder 1, $\mathbf{a}_1$ is its axial direction and $r_2$ is the radius of cylinder 2.

## 7.4 Examples of constraints

The set of constraints presented in the last section can be used to specify the relationship between different CSG primitives. Here, we will outline the details of constraint specification for a flanged T junction and a flanged curve (Figure 7.2).

### 7.4.1 Flanged T-Junction

This CSG object consists of a union of 5 cylinders. It has the following geometric constraints:

- Parameter constraints
    1. The radii of cylinders 1 and 2 are the same ($r_1 = r_2$).
    2. The radii of cylinders 3, 4, & 5 are equal and always greater than those of cylinder 1 and 2 ($r_3 = r_4 = r_5 \quad \& \quad r_3 > r_1$)
    3. The length of cylinder 2 is half that of cylinder 1. ($l_2 = \frac{1}{2} l_1$).
    4. The lengths of cylinders 3, 4, & 5 are equal and are less than the lengths of cylinders 1 and 2 ($l_3 = l_4 = l_5 \quad \& \quad l_3 < l_2 < l_1$).
    5. All the radii and the lengths are within given limits which should be positive ($r_{min} < r < r_{max} \quad \& \quad l_{min} < l < l_{max}$).
- Direction constraints
    1. Cylinders 1 & 2 are perpendicular ($\mathbf{a}_1 \perp \mathbf{a}_2 \Rightarrow \cos^{-1}(\mathbf{a}_1 \cdot \mathbf{a}_2) = \frac{\pi}{2}$).
    2. Cylinders 2 and 3 have the same direction ($\cos^{-1}(\mathbf{a}_2 \cdot \mathbf{a}_3) = 0$).
    3. Cylinders 1, 4 & 5 have the same direction ($\cos^{-1}(\mathbf{a}_1 \cdot \mathbf{a}_4) = \cos^{-1}(\mathbf{a}_1 \cdot \mathbf{a}_5) = 0$).
- Translation constraints
    1. Cylinder 2 can slide along cylinder 1. $\mathbf{t}_2 = \mathbf{t}_1 + c_1 \mathbf{a}_1$
    2. Cylinders 4 & 5 are placed on the two ends of cylinder 1 ($\mathbf{t}_4 = \mathbf{t}_1 \quad \& \quad \mathbf{t}_5 = \mathbf{t}_1 + l_1 \mathbf{a}_1$).
    3. Cylinder 3 is placed at the end of cylinder 2 ($\mathbf{t}_3 = \mathbf{t}_2 + l_2 \mathbf{a}_2$).
    4. Cylinder 2 has bounded translation constraint with cylinder 1 ($\mathbf{t}_1 + r_2 < \mathbf{t}_2 < \mathbf{t}_1 + l_1 - r_2$).

### 7.4.2 Flanged curved pipe

- Parameter Constraints

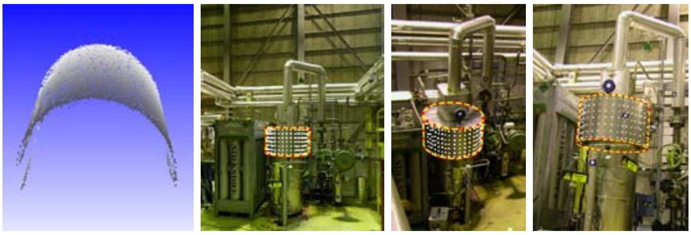
1. The radii and the lengths of cylinders 2 and 3 are equal ($r_2 = r_3 \quad \& \quad l_2 = l_3$).
2. The radius of the curve is less than the radii of the cylinders ($r_1 < r_2$)
3. The major radius of the curve is at least twice that of the minor radius ($2r_1 < R_1$).
4. All the radii and the lengths are within given positive limits ($r_{min} < r < r_{max} \quad \& \quad l_{min} < l < l_{max}$).

- Rotation constraints: The tangential direction of the curve and the axial directions of cylinders 2 and 3 should be the same. That gives us the following two equations:
  1. $\cos^{-1}\left(\mathbf{a}_2 \cdot \mathbf{R}_t \begin{pmatrix}-\sin\alpha_a & \cos\alpha_a & 0\end{pmatrix}^T\right) = 0$
  2. $\cos^{-1}\left(\mathbf{a}_3 \cdot \mathbf{R}_t \begin{pmatrix}-\sin\alpha_b & \cos\alpha_b & 0\end{pmatrix}^T\right) = 0$ Where $\alpha_a$ is the starting angle and $\alpha_b$ is the ending angle of the curve and $\mathbf{R}_t$ is its rotation matrix.
- Translation Constraints
  1. The position of cylinder 2 is at the start of the curve ($\mathbf{t}_2 = \mathbf{t}_1$).
  2. The position of cylinder 3 is at the end of the curve i.e. $\mathbf{t}_3 = \mathbf{t}_1 + R_1\mathbf{R}_t \begin{pmatrix}\cos\alpha & sin\alpha & 0\end{pmatrix}^{\mathrm{T}}$. $\mathbf{R}_t$ is the rotation matrix, $R_1$ is the major radius of the torus and $\alpha$ is the angular extent of the toroidal curve ($\alpha = \frac{\pi}{2}$ in Figure 7.2(b)).

## 7.5 Fitting Experiments

As it was said in the introduction, images and point clouds provide complementary sources of information, and by their combination we can expect better estimation accuracy. Edges of the object, where point clouds are usually noisy, are captured with better quality in the images. Additionally, for the fitting of bounded objects point clouds do not contain enough information for the estimation of the bounds. In contrast, by providing the full edge outline images fix the bounds. For example in the case of a cylinder usually the closing lids on both sides are not scanned either because they are not visible due to the connections with the other surrounding objects, or because it is not convenient to place the scanner in a position where the lids are visible. As a result we expect the length of the cylinder to be poorly determined by such a point cloud. In contrast the measurements in the image provide points on the edges and their inclusion improves the quality of the length estimation.

To demonstrate the complementary nature of the images and the point clouds we will do some fitting experiments on two test objects. Each object will be fitted three times, first using only the point cloud, then using only the image measurements and finally using both data sources simultaneously. The point clouds we will use were captured using a Cyrax HDS 2500 scanner. As suggested by Leica (Leica



(a) (b) (c) (d)

**Figure 7.5:** *Experiment for cylinder fitting (a) The point cloud (b-d) Images used for reconstruction*

HDS, 2005), we assume a standard deviation of 5mm for each point. The images were captured using a Nikon CoolPix camera having a resolution of 5 mega pixels and using a fixed focal length of 7.34 mm. The standard deviation for image measurements is taken to be 1 pixel.

| | Parameter | Image | | | Point Cloud | Both |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | | |
| 1 | X (mm) | 52.269 | 10.326 | 9.923 | 0.838 | 0.762 |
| 2 | Y (mm) | 163.45 | 14.168 | 13.629 | 1.335 | 0.862 |
| 3 | Z (mm) | 12.740 | 3.489 | 3.467 | 119.90 | 1.654 |
| 4 | t0 | 0.056 | 1.0E-2 | 1.0E-2 | 2.28E-3 | 2.0E-3 |
| 5 | t1 | 0.065 | 2.1E-2 | 2.1E-2 | 3.96E-3 | 2.9E-3 |
| 6 | t2 | 5.283 | 2.538 | 2.534 | 0.359 | 0.237 |
| 7 | Length (mm) | 10.591 | 3.121 | 3.120 | $\infty$ | 3.120 |
| 8 | Radius (mm) | 169.57 | 16.166 | 15.522 | 0.634 | 0.565 |

**Table 7.1:** *Standard deviations for Cylinder fitting experiment*

### 7.5.1 Cylinder fitting

The arrangement we used for the first experiment is shown in Figure 7.5. A cylinder is scanned from the front, and images are taken from three different positions. A cylinder is represented by 8 parameters, 3 for the position, 3 for the axis, one for the radius and one for the length. In Table 7.1 we see the standard deviations obtained for different parameters by doing fitting to the point clouds, the images and to a combination of both. For images we did fitting three times using one, two and three images. When doing the combined fitting the point cloud was used along with all of the three images. As expected, in the case of using only the point cloud the length of cylinder is not well-determined because in the absence of points on upper and lower lids there is not enough information in the point cloud for its determination. This results in a standard deviation of $\infty$ for the length, meaning it is yet undetermined (Table 7.1).

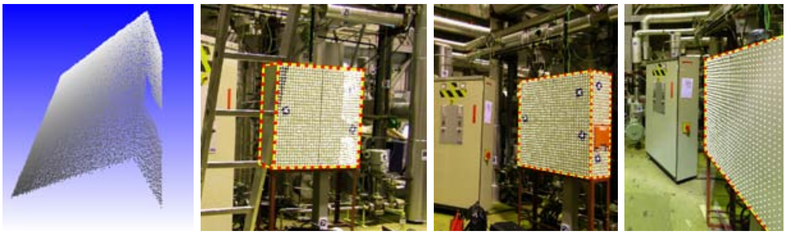

(a) (b) (c) (d)

**Figure 7.6:** *Experiment for box fitting (a) The point cloud (b-d) Images used for reconstruction*

As the z-axis of the global coordinate system is aligned with the axial direction of the cylinder there is a very high correlation between the length and the position in z. This results in a weak estimation of the z-position compared to the estimation of x and y position. But if we combine the point cloud with the measurements from the images (Table 7.1, column "Both") the situation improves dramatically as the edges in the images provide enough information for the estimation of the length and the resulting standard deviations are much lower, indicating much better estimation quality.

As expected as we use more images the standard deviation of parameter estimation goes down and the estimation quality improves. It also shows that even a single image in combination with a good scan can lead to a significant improvement in the estimation of those parameters which are not well-determined from the point cloud.

### 7.5.2 Box Fitting

| | Parameter | Images | | | Point Cloud | Both |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | | |
| 1 | X (mm) | 2.106 | 1.179 | 0.685 | 3.079 | 0.649 |
| 2 | Y (mm) | 2.243 | 1.129 | 0.261 | 0.550 | 0.161 |
| 3 | Z (mm) | 1.669 | 0.760 | 0.338 | 389.84 | 0.314 |
| 4 | q0 | 7.8e-1 | 2.6e-1 | 5.3e-2 | 3.97e-2 | 3.0e-2 |
| 5 | q1 | 1.4e-3 | 5.0e-4 | 1.3e-4 | 2.40e-4 | 1.0e-4 |
| 6 | q2 | 7.8e-4 | 3.0e-4 | 6.0e-5 | 5.20E-4 | 5.0e-5 |
| 7 | q3 | 4.3e-3 | 1.6e-3 | 3.4e-4 | 3.40E-4 | 2.0e-4 |
| 8 | X size (mm) | 2.855 | 1.023 | 0.689 | 2.890 | 0.661 |
| 9 | Y size (mm) | 9.836 | 2.696 | 0.627 | $\infty$ | 0.627 |
| 10 | Z size (mm) | 2.309 | 1.161 | 0.349 | $\infty$ | 0.348 |

**Table 7.2:** *Standard deviations for Box fitting experiment*

The second example is that of a box, with only two of its faces fully scanned.

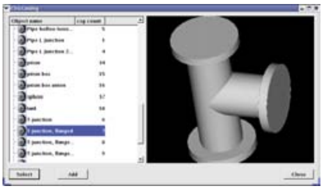


**Figure 7.7:** *Catalog of CSG objects used for modeling. Each objects contains the CSG tree along with the internal equality and inequality constraints.net*

Additionally, three images are taken from different positions (Figure 7.6). The box has 10 parameters, 3 for the position, 4 for the rotation, and 3 for the sizes.

In the absence of points on all faces of the box, it is not possible to reliably determine the size parameters of the box. This is what we see in the standard deviations resulting from fitting using only point clouds (Table 7.2). The standard deviation for y and z sizes are $\infty$ meaning that they could not be estimated.[1] The standard deviation for x size is low only because of the coordinate system chosen for the box, which has its origin in the left corner. This fixes the position of the right side and thus the x size is also determined. Due to the high correlation between the position and the size along the z-axis, its estimation is also bad.

The results of fitting the box to images and the points cloud are given in Table 7.2. Once again we see from the last column that the inclusion of image measurements leads to a much better estimation of the size and the position parameters.

Both of these examples prove our thesis, that although point clouds contain direct 3D information, which is very useful for automatic object recognition, the final estimation should use a combination of both data sources. The inclusion of images compensates for the missing or noisy information from the point clouds and produces parameter estimates with better quality.

## 7.6 Results

We applied the presented methodology for 3D reconstruction to an industrial site shown in Figure 7.8. Seven scans were made using a Cyrax laser scanner. Each scan consisted of one million points with a standard deviation of 5mm. Additionally 15 images were taken from different positions and used during fitting.

The process of combining automatically detected cylinders and planes to full CSG objects as well as the process of adding measurements to images was done manually. Once we have image measurements as well as segmented points clouds, we proceed with the CSG fitting using constraints. Commonly found CSG objects were kept in a catalog, that contained the CSG tree as well as the internal equality and inequality constraints (Figure 7.7). The use of such a catalog provides a flexible way to encode domain specific information in the form of geometry with

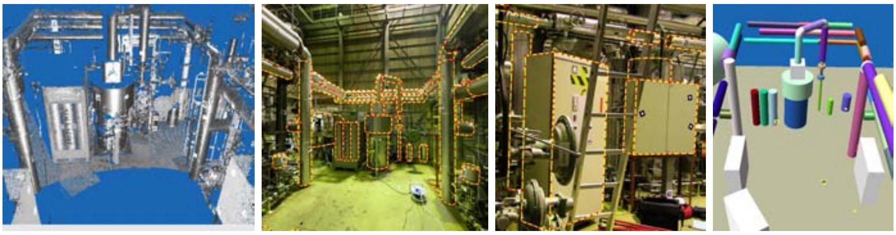
---

[1] It should be noted that it is only our choice of the coordinate system that leads to this effects arising only for z. In a rotated coordinate system all three x, y and z sizes will have very poor estimation quality (high standard deviation).

constraints.

By comparing the point cloud in Figure 7.8(a) with the corresponding images in Figure 7.8(b–d) we see that although the point cloud captures the geometry of objects quite well, the information about their bounds is either absent or is very weak. In comparison, the images provide excellent complementary source as they contain information about the bounds in the form of object edges. Thus, by combining both sources of data a better estimation quality was obtained.

The CSG fitting procedure minimizes the sum of square of the distances of point cloud from the model surface and sum of square of the image measurement distance from the back projected edges of the model, while estimating the pose and shape parameters of the CSG objects. This process uses both point cloud and image measurements in one combined estimation procedure and gives final results as a CSG model. The final model is shown in Figure 7.8(d).



(a) (b) (c) (d)

**Figure 7.8:** *Modeling of an industrial installation (a) The registered point cloud (b-c) Images with back-projected model and measured points on the edges (d) The final 3D model*

## 7.7 Conclusions

We have presented a modeling technique for fitting CAD models described as CSG trees to point clouds and the measurements in images. While the point clouds are excellent for automatic object recognition, the improvement in the standard deviation of the estimated parameters clearly shows that images have a complementary role because they provide better information on the edges and help fix the bounds of the models; a job point clouds cannot do very well. We presented a set of constraints which can be used to specify and enforce geometric relationships between sub-parts of one or more CSG trees. The constrained problem was solved using Lagrange multipliers, and compared to the previously presented method of explicit constraints in Chapter 4 this provided a more flexible and general solution. The presented methods were applied to data sets from an industrial site and the results show the improvements obtained by including constraints and image information.

## Chapter 8

# Conclusions

> Reasoning draws a conclusion, but does not make the conclusion certain, unless the mind discovers it by the path of experience.
>
> Roger Bacon (1214-1294)

> The power of generalizing ideas, of drawing comprehensive conclusions from individual observations, is the only acquirement, for an immortal being, that really deserves the name of knowledge.
>
> Mary Wollstonecraft (1759-1797)

In this thesis we have addressed different aspects of 3d reconstruction using point clouds and images. By restricting our focus to only industrial installations we were able to exploit the domain specific constraints and apriori information for formulating automatic and semiautomatic methods for segmentation, object recognition, registration and model fitting. The following are some of the achievements of our work:

## 8.1 Achievements

**Segmentation** We developed a fast point cloud segmentation method that uses only the surface normals which can be reliably estimated from often noisy point clouds by fitting planes to the neighbors of each point. The algorithm is based on region growing and groups together the points belonging to smooth regions. The *smoothness constraint* is controlled with only two thresholds, which have intuitive and predictable effects. The thresholds can be easily changed to get different trade-offs between under- and over-segmentation. The segmentation is a pre-processing step for the next stage of object recognition. As a result under-segmentation is not a serious problem because the stage of object recognition based on the Hough transform can easily recover from its effects.

**Object recognition** In industrial environments planes and cylinders are two of the most commonly found objects. By using a combination of segmentation and the Hough transform we were able to get a high success rate for the detection of planar and cylindrical objects in the cluttered industrial environments. An efficient two step approach was formulated to tackle the problems of space and computational complexity arising from a direct application of the Hough transform for cylinder detection in point clouds. This two step approach estimates cylinder orientation in the first step by using a 2D Hough space. The second step of position and radius estimation requires a 3D Hough space. This sequential approach makes the complexity of the problem manageable. Moreover, by localizing the serach through segmentation we were able to combine the efficiency of local processing with the advantages of global search obtained from the Hough transform.

**Model fitting through least squares** To estimate the parameters of the objects detected through the Hough transform and to get quality measures for each parameter, fitting methods based on nonlinear least squares were developed. Moreover, to avoid the stage of manual editing, commonly used for B-rep based fitting approaches, we developed the CSG fitting techniques. Three different methods for calculating the distance of a point from a given CSG object were developed and compared. Initially the internal constraints, which specify the relationships between the subparts of a CSG tree, were implemented by parameter elimination. This made possible the use of unconstrained least squares methods for solving this constrained problem.

**Registration and integrated adjustment** A new paradigm of object based registration was presented that made use of the available structure present in the industrial scenes to estimate the inter-scan transformation parameters. The corresponding objects in two or more scans were used to register them in one coordinate system. We presented constraints for an efficient and automatic search for the corresponding objects in a given pair of scans. These constraints expoited the partial information from the match decisions of the past to filter the possiblities for the future matches. By first detecting the planes and the cylinders in each scans and then searching for the corresponding objects the whole process of registration can be automated. Moreover, the model-based registration framework allows global registration through integrated adjustment, that simultaneously estimates the transformation parameters of the scans and the shape parameters of the corresponding objects. We compared the results of our approach with ICP and showed that our approach leads to better estimation quality because of the global registration.

**Constrained CSG fitting including images** To make the process of constraint specification and enforcement between sub-parts of different CSG objects more flexible, we presented fitting techniques based on Lagrange multipliers. Mathematical formulation of different commonly found equality and inequality constraints for CSG models were given. We showed by examples that for high accuracy as-built reconstruction images still provide very valuable information especially on the edges where point clouds are usually quite noisy. We extended the procedure of CSG fitting for simultaneous use of the point clouds and the images. This CSG fitting to the images minimizes the sum of the distances between the measurements on the edges in the captured images from the back projected contours of the model being fitted. By combining both the point clouds and the images in one estimation procedure we were able to get higher estimation quality.

## 8.2 Directions for future work

Although in this thesis we have tried to address most of the steps involved in 3d industrial reconstruction, many problems still remain unsolved and can benefit from future research. The following are some of the directions for future work:

- Recently the use of off the shelf GPS and digital compass have been proposed and tested for getting good approximate values for automatic registration, see for example Böhm and Haala (2005); Schuhmacher and Böhm (2005). GPS can only provide information about the position of the scans and getting reliable information about their orientation is still quite problemetic. The quality of the positional information would become even better with the next generation of global navigation satellite systems (GALILEO, 2005). By combining this approximate information about the position in a constraint based fromework for correspondence search the process of automatic registration can be made more robust. As planes make more than 80% of the urban scenes this strategy can also be very useful for mobile mapping.

- With the amount of data the new generation of laser scanners are producing it is becoming challenging to store and process it. Techniques need to be developed that exploit the local information to compress the data. For example real-time segmentation into planar areas and their use in a predictive-coding framework can both result in better compression and extraction of information that can later be used by other processing stages.

- The CSG fitting methods we presented made point-wise decisions. Each point was iteratively assigned to one of the surfaces making the CSG model. As segmentation provides grouped information, its combination with CSG fitting can result in better and robust fitting methods. The same reasoning applies to the use of images. By doing edge based decisions on a segmented edge map, the process of using image measurements can be improved.

- Currently the step of specifying the relationships between simple surfaces and CSG models is done manually. More research is needed to develop a process of ‘automatic refinement’ which can make a sensible model by grouping the fitted pieces together, and assign them to the best possible CSG model for fitting. Different approaches can be possible to address this problem. By including more domain specific information the performance of such systems can be improved. A heuristic or rule-based search on the automatically detected simple objects like planes and cylinders is one of the possible ways to develop an automatic or semi-automatic method for this step. Such a method if combined with a catalog of commonly found objects in the industrial environment under consideration should lead to an even higher level of automation.

- There is a need to combine the 3d reconstruction methods with the 3d design software packages. By using the same CSG representation during both the design and the reconstruction the user gets two main advantages. Firstly, the set of constraints used during design are also enforced during reconstruction. Secondly, the CAD model resulting from the CSG fitting is directly useable without any conversion or human interpretation because the same format is used for both processes.

- Most of the new laser scanner are coming with the option of acquiring registered imagery from the same viewpoint. This should provide a strong possibility to use complementary image information automatically. The objects which are fitted in the point cloud can be automatically back-projected in the registered images, and by doing rule-based analysis on the image edges, the image measurements can be automatically or semi-automatically added. This will remove the laborious process of adding image measurments manually and should lead to a more wide-spread use of the readily available image information for improving the quality of 3d reconstruction.

# Appendix A

# Uniform sampling of the orientation space

The shape that results from plotting all possible unit vectors in $\mathbb{R}^3$ is called the Gaussian sphere (Carmo, 1976). It is a map of all orientations possible in 3D. As the Hough transform works on a discrete representation of the parameter space, we need a sampling method that places points on the sphere, so that each point represents an equal area on the surface. This is essential for an unbiased Hough transform, otherwise the results will be biased for the samples whose cells represent bigger areas, and thus collect more votes.

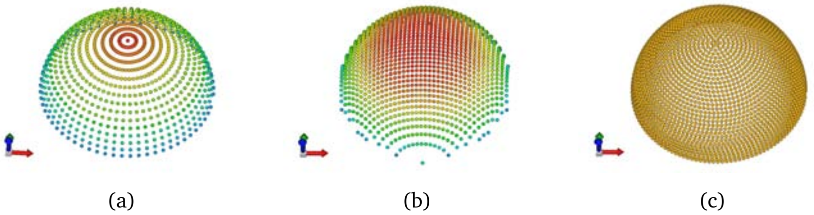

(a) (b) (c)

**Figure A.1:** *Comparison of different methods for sampling the orientation space in $\mathbb{R}^3$ (a) Spherical sampling (b) Cartesian sampling (c) Approximate uniform sampling*

As noted in (Hoppe et al., 1992) linear sampling of $\theta$ and $\phi$ spherical coordinates, which we will call *Spherical sampling*, results in non-uniform area cells. This makes the cells near the Z-axis much smaller than the ones close to the $xy$ plane (Figure A.1(a)).

Another option is to uniformly sample $x$ and $y$, and then use explicit equation of the unit sphere for calculating $z$ :

$$z = \pm\sqrt{1 - x^2 - y^2} \tag{A.1}$$

We name this sampling scheme *Cartesian sampling*. As is clear from Figure A.1(b) the cartesian sampling is better than the spherical sampling, but here also the cells get bigger as we reach the $xy$ plane.

The exact solution for uniform sampling of the sphere with equal area cells has no closed-form solution but requires an iterative refinement of some initial approximate distribution (Rusin, 2004). For the Hough transform an approximate solution that is also able to map back and forth from the accumulator indices to the original parameter values is desirable. An iterative procedure would not allow us to do this mapping efficiently and will slow down the Hough transform. Next, we present such an approximate sampling scheme using the concepts from (Lutton et al., 1994; Rusin, 2004).

The basic idea for getting a uniform sampling of the Gaussian sphere is to sample $\phi$ uniformly and change the sampling density along $\theta$ adaptively. A constant value of $\phi$ on the sphere results in a circular curve, whose radius is a function of $\phi$. Consequently, for a uniform sampling the number of samples along $\theta$ becomes a function of $\phi$.

Let the number of samples in $\phi$ be $n_\phi$, then for a given value of $\phi_i$ the number of samples in $\theta$, $n_{\theta i}$, is given by:

$$n_{\theta i} = f(\sin \phi_i) = 2n_\phi \sin \phi_i + 1 \tag{A.2}$$

The factor of 2 in Equation A.2 appears because $\theta_{range} = 2\phi_{range}$. We name this sampling method *Approximate uniform sampling*, and its results are shown in Figure A.1(c).

Approximate uniform sampling has the attractive property, that knowing $\theta_{index}$ and $\phi_{index}$ in the Hough space, we can easily recover the values of $\theta_o$ and $\phi_o$ in the parameter space using the following formulas:

$$\phi_o = \phi_{index}\phi_{step} \tag{A.3}$$

$$\theta_{step} = \frac{\pi}{2sin\phi_o} \tag{A.4}$$

$$\theta_o = \theta_{step}\theta_{index} \tag{A.5}$$

This sampling method is used for discretization of the orientation space for the plane normals in the plane detection (Section 3.2) and for the cylinder axes during the cylinder detection (Section 3.3).

# Bibliography

Ahn, S. J., Rauh, W., Cho, H. S. and Warnecke, H.-J., 2002. Orthogonal distance fitting of implicit curves and surfaces. PAMI 24(5), pp. 620–638.

Alter, T. D. and Jacobs, D. W., 1998. Uncertainty propagation in model-based recognition. IJCV 27(2), pp. 127–159.

Amano, T., Hiura, S., Yamaguchi, A. and Inokuchi, S., 1996. Eigen space approach for a pose detection with range images. In: ICPR '96, Vol. 1, IEEE Computer Society, Washington, DC, USA, p. 622.

Arman, F. and Aggarwal, J. K., 1993. Model-based object recognition in dense-range images - a review. ACM Computing Surveys 25(1), pp. 5–43.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. and Wu, A. Y., 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of ACM 45(6), pp. 891–923.

Atkinson, K. B. (ed.), 1996. Close Range Photogrammetry and Machine Vision. Engineering and Science, Whittles Publishing, Scotland, UK. 12 chapters, 371 pages.

AVEVA PDMS, 2005. PDMS–Plant Design and Management System (AVEVA Engineering IT). `http://www.aveva.com/library/datasheets/pdf/vpd_pdms.pdf`.

Ballard, D., 1981. Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition 13(2), pp. 111–122.

Bellman, R., 1961. Adaptive control processes: A guided tour. Princeton University Press., Princeton, PA.

Bellon, O. R. P., Direne, A. I. and Silva, L., 1999. Edge detection to guide range image segmentation by clustering techniques. In: International Conference on Image Processing (ICIP '99), Kobe, Japan, pp. 725–729.

Bennamoun, M. and Mamic, G. J., 2002. Object recognition: fundamentals and case studies. Springer-Verlag New York, Inc.

Bergen, J. R. and Schweitzer, H., 1991. A probabilistic algorithm for computing hough transforms. Journal of Algorithms 12(4), pp. 639–656.

Bertsekas, D. P., 1995. Nonlinear Programming. Athena Scientific, Belmont, MA.

Besl, P. J. and Jain, R. C., 1985. Three-dimensional object recognition. ACM Computing Surveys 17(1), pp. 75–145.

Besl, P. J. and Jain, R. C., 1988a. Segmentation through variable-order surface fitting. PAMI 10(2), pp. 167–192.

Besl, P. J. and Jain, R. C., 1988b. Segmentation through variable-order surface fitting. PAMI 10(2), pp. 167–192.

Besl, P. J. and McKay, N. D., 1992. A method for registration of 3-d shapes. PAMI 14(2), pp. 239–256.

Bhanu, B., Lee, S., Ho, C. C. and Henderson, T., 1986. Range data processing: representation of surfaces by edges. In: Proceedings of the Eighth International Conference on Pattern Recognition, pp. 236–238.

Bishop, C. M., 1996. Neural networks for pattern recognition. Oxford University Press, Oxford, UK, UK.

Björck, Å., 1996. Numerical Methods for Least Squares Problems. SIAM, Philadelphia.

Blais, F., 2004. A review of 20 years of range sensor development. Journal of Electronic Imaging 13(1), pp. 231–240.

Böehler, W., Bordas, M. and A., M., 2003. Investigating laser scanner accuracy. In: Proc. CIPA XIXth Int. Symposium, Antalya, Turkey., pp. 696–702.

Böhler, W., Heinz, G., Marbs, A. and Siebold, M., 2002. 3d scanning software - an introduction. In: International Workshop on Scanning for Cultural Heritage Recording (CIPA 2002), Corfu, Greece, pp. 9–13.

Böhm, J. and Haala, N., 2005. Efficient integration of aerial and terrestrial laser data for virtual city modeling using lasermaps. In: G. Vosselman and C. Brenner (eds), Proceedings of the ISPRS Workshop, Laser scanning 2005, Vol. XXXVI, ISPRS, Enschede, the Netherlands, pp. 192–195.

Bolles, R. C. and Fischler, M. A., 1981. A ransac-based approach to model fitting and its application to finding cylinders in range data. In: Proc. of the 7th International Joint Conference on Artificial Intelligence, Vancouver, Canada, pp. 637–643.

Bookstein, F., 1979. Fitting conic sections to scattered data. Computer Graphics and Image Processing 9(1), pp. 56–71.

Borg, K. R. and Bielajew, A. F., 1995. Quadplot: A programme to plot quadric surfaces. Technical Report NRCC/PIRS-0491, Institute for National Measurement Standards, National Research Council of Canada, Ottawa, Canada.

Bowyer, K., Sallam, M., Eggert, D. and Stewman, J., 1993. Computing the generalized aspect graph for objects with moving parts. PAMI 15(6), pp. 605–610.

Box, G. E. P. and Draper, N. R., 1986. Empirical model-building and response surface. John Wiley & Sons, Inc., New York, NY, USA.

Campbell, R. J. and Flynn, P. J., 1999. Eigenshapes for 3d object recognition in range data. In: CVPR, pp. 2505–2510.

Carmo, M. P. D., 1976. Differential Geometry of Curves and Surfaces. Prentice-Hall.

Chaperon, T. and Goulette, F., 2001. Extracting cylinders in full 3d data using a random sampling method and the gaussian image. In: Proc. Vision, Modelling and Visualisation, University of Stuttgart, Germany, pp. 35–42.

Chapman, D., Deacon, A. and Lucas-Brown, J., 2004. An omnidirectional imaging system for the reverse engineering of industrial facilities. In: H. G. Maas and D. Schneider (eds), Panoramic Photogrammetry Workshop, IAPRS, Vol. 34number XXX, Dresden, Germany.

Chen, Y. and Medioni, G. G., 1992. Object modeling by registration of multiple range images. IVC 10(3), pp. 145–155.

Clark, J. and Robson, S., 2004. Accuracy of measurements made with a cyrax 2500 laser scanner. In: ISPRS XX Congress, IAPRS, Vol. XXXV, B3, Istanbul, Turkey.

Curless, B. and Levoy, M., 1996. A volumetric method for building complex models from range images. In: SIGGRAPH, pp. 303–312.

Cyr, C. M. and Kimia, B. B., 2004. A similarity-based aspect-graph approach to 3d object recognition. IJCV 57(1), pp. 5–22.

Daniilidis, K. and Bayro-Corrochano, E., 1996. The dual quaternion approach to hand–eye calibration. In: Proc. 13th Int. Conf. on Pattern Recognition, Vol. A, Vienna, ICPR, pp. 318–322.

Dennis, Jr., J. and Schnabel, R. B., 1996. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM. [ Edital Universal, 2001 ].

Dijkman, S. and Heuvel, F. A. v. d., 2000. Semi automatic registration of laser scanner data. In: Proceedings of the XIXth congress of ISPRS, Vol. XXXIII, part B 5/1, pp. 215–221.

Dorai, C., Weng, J. and Jain, A., 1994. Optimal registration of multiple range views. In: ICPR94, pp. A:569–571.

Duda, R. O. and Hart, P. E., 1972. Use of the hough transformation to detect lines and curves in pictures. Communincations of ACM 15(1), pp. 11–15.

Eberly, D., 2001. 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. Morgan Kaufmann/Academic Press.

Eggert, D. W., Bowyer, K. W., Dyer, C. R., Christensen, H. I. and Goldgof, D. B., 1993. The scale space aspect graph. PAMI 15(11), pp. 1114–1130.

Eggert, D. W., Fitzgibbon, A. W. and Fisher, R. B., 1998a. Simultaneous registration of multiple range views for use in reverse engineering of cad models. CVIU 69(3), pp. 253–272.

Eggert, D. W., Fitzgibbon, A. W. and Fisher, R. B., 1998b. Simultaneous registration of multiple range views for use in reverse engineering of cad models. CVIU 69(3), pp. 253–272.

El-Hakim, S., Beraldin, J.-A., Picard, M. and Godin, G., 2004. Detailed 3d reconstruction of large-scale heritage sites with integrated techniques. In: IEEE Computer Graphics and Applications, Vol. 24number 3, pp. 21–29.

Eriksson, J. and Wedin, P.-Å., 2004. Truncated gauss-newton algorithms for ill-conditioned nonlinear least squares problems. Optimization Methods and Software 19(6), pp. 721 – 737.

Ermes, P., Heuvel, F. A. v. d. and Vosselman, G., 1999. A photogrammetric measurement method using csg models. In: IAPRS, Vol. 32, part 5/W11, pp. 36–42.

Faber, P. and Fisher, R. B., 2001a. A buyer's guide to euclidean elliptical cylindrical and conical surface fitting. In: T. F. Cootes and C. Taylor (eds), BMVC, British Machine Vision Association, Manchester, UK.

Faber, P. and Fisher, R. B., 2001b. Pros and cons of euclidean fitting. In: B. Radig and S. Florczyk (eds), DAGM Symposium, Lecture Notes in Computer Science, Vol. 2191, Springer, Munich, Germany, pp. 414–420.

Farin, G., 2002. Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Fischer, A. and Steinhage, V., 1997. On the computation of visual events in aspect graph generation. In: Mustererkennung 1997, 19. DAGM-Symposium, Springer-Verlag, London, UK, pp. 156–163.

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of ACM 24(6), pp. 381–395.

Fischler, M. A. and Bolles, R. C., 1987. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: M. A. Fischler and O. Firschein (eds), Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, Kaufmann, Los Altos, CA., pp. 726–740.

Fisher, R. B., 2002. Applying knowledge to reverse engineering problems. In: Geometric Modeling and Processing (GMP 2002), Theory and Applications,, Wako, Saitama, Japan, pp. 149–155.

Flynn, P. J. and Jain, A. K., 1988. On reliable curvature estimation. Technical Report MSU-CPS-88-01, Department of Computer Science, Michigan State University, East Lansing, Michigan. also available as technical report PRIP-88-1.

Foley, J. D., van Dam, A., Feiner, S. K. and Hughes, J. F., 1990. Computer graphics: principles and practice (2nd ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Forsey, D. R. and Bartels, R. H., 1995. Surface fitting with hierarchical splines. ACM Transactions on Graphics 14(2), pp. 134–161.

Fraleigh, J. B. and Beauregard, R. A., 1995. Linear Algebra. Third edn. Principal axis theorem - sec 8.1.

Fugro, 2006. Fugro Inpark BV, The Netherlands. `http://www.fugro-inpark.nl/information/aboutus/aboutus.asp`.

Fukunaga, K., 1990. Introduction to statistical pattern recognition (2nd ed.). Academic Press Professional, Inc., San Diego, CA, USA.

GALILEO, 2005. GALILEO – European Satellite Navigation System. `http://europa.eu.int/comm/dgs/energy_transport/galileo/index_en.htm`.

Gigus, Z. and Malik, J., 1988. Computing the aspect graph for line drawings of polyhedral objects. Technical report, Berkeley, CA, USA.

Golub, G. H. and Loan, C. F. V., 1991. Matrix Computation. John Hopkins University Press. Second Edition.

Goodman, J. E. and O'Rourke, J., 1997. Handbook of discrete and computational geometry. CRC Press, Inc., Boca Raton, FL, USA.

Greenspan, M., 1998. The sample tree: A sequential hypothesis testing approach to 3d object recognition. In: CVPR '98, IEEE Computer Society, Washington, DC, USA, p. 772.

Greenspan, M. and Boulanger, P., 1999. Efficient and reliable template set matching for 3d object recognition. In: 3DIM99: 3D Imaging and Modelling, pp. 230–239.

Grimson, W. E. L., Huttenlocher, D. P. and Jacobs, D. W., 1994. A study of affine matching with bounded sensor error. IJCV 13(1), pp. 7–32.

Guehring, J., 2001. Reliable 3d surface acquisition, registration and validation using statistical error models. In: Third International Conference on 3-D Digital Imaging and Modeling (3DIM '01), pp. 224–231.

Gupta, A. and Bajcsy, R., 1993. Volumetric segmentation of range images of 3d objects using superquadric models. CVGIP 58(3), pp. 302–326.

Helfrich, H.-P. and Zwick, D., 1993. A trust region method for implicit orthogonal distance regression. Numerical Algorithms 5(4), pp. 535–545.

Heuvel, F. A. v. d., 1999. A line-photogrammetric mathematical model for the reconstruction of polyhedral objects. In: S. F. El-Hakim (ed.), Videometrics VI, Proceedings of SPIE, Vol. 3641, pp. 60–71.

Heuvel, F. A. v. d., 2003. Rules of thumb for efficient recording of an industrial site with optical and range data. Technical report, Section of Photogrammetry and Remote Sensing, Delft University of Technology, The Netherlands.

Hoover, A., Jean-Baptiste, G., Goldgof, D. and Bowyer, K., 1995. A methodology for evaluating range image segmenation techniques. In: Procs. International Symposium on Computer Vision, Florida, USA.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K. K., Eggert, D. W., Fitzgibbon, A. W. and Fisher, R. B., 1996. An experimental comparison of range image segmentation algorithms. PAMI 18(7), pp. 673–689.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., 1992. Surface reconstruction from unorganized points. In: Proceedings of ACM SIGGRAPH ’92, ACM Press, New York, NY, USA, pp. 71–78.

Horn, B. K. P., 1987. Closed-form solution of absolute orientation using unit quaternions. JOSA 4(4), pp. 629–642.

Horn, B. K. P., Hilden, H. and Negahdaripour, S., 1988. Closed-form solution of absolute orientation using orthonormal matrices. JOSA 5(7), pp. 1127–1135.

Hough, P. V. C., 1962. Method, and means for recognizing complex patterns, U. S. Patent 3069654.

Huttenlocher, D. P. and Ullman, S., 1990. Recognizing solid objects by alignment with an image. IJCV 5(2), pp. 195–212.

Intergraph PDS, 2005. PDS–Plant Design System (Intergraph Process, Power & Marine). `http://http://ppm.intergraph.com/pds/`.

iQvolution, 2005. FARO iQvolution 3D laser scanners. `http://www.iqvolution.com/en/Products/Laserscanners.php`.

Jahne, B., Geissler, P. and Haussecker, H. (eds), 1999. Handbook of Computer Vision and Applications. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Jain, A. K. and Dorai, C., 2000. 3d object recognition: Representation and matching. Statistics and Computing 10(2), pp. 167–182.

Jiang, X., Bowyer, K. W., Morioka, Y., Hiura, S., Sato, K., Inokuchi, S., Bock, M., Guerra, C., Loke, R. E. and du Buf, J. M. H., 2000. Some further results of experimental comparison of range image segmentation algorithms. In: ICPR '2000, Vol. 4, IEEE Computer Society, Washington, DC, USA, p. 4877.

Jiang, X., Hoover, A., Jean-Baptiste, G., Goldgof, D., Boywer, K. and Bunke, H., 1995. A methodology for evaluating edge detection techniques for range images. In: Proc. Asian Conf. Computer Vision, pp. 415–419.

Jiang, X. Y., Bunke, H. and Meier, U., 1996. Fast range image segmentation using high-level segmentation primitives. In: Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), IEEE Computer Society, Washington, DC, USA, p. 83.

Johnson, A. and Hebert, M., 1997. Surface registration by matching oriented points. In: International Conference on Recent Advances in 3-D Digital Imaging and Modeling, pp. 121–128.

Kalviainen, H., Hirvonen, P., Xu, L. and Oja, E., 1994. Comparisons of probabilistic and non-probabilistic hough transforms. In: ECCV '94, Springer-Verlag, London, UK, pp. 351–360.

Khalifa, I., Moussa, M. and Kamel, M., 2003. Range image segmentation using local approximation of scan lines with application to cad model acquisition. Machine Vision Applications 13(5-6), pp. 263–274.

Kimme, C., Ballard, D. and Sklansky, J., 1975. Finding circles by an array of accumulators. Communications of ACM 18(2), pp. 120–122.

Ko, K. H., Maekawa, T. and Patrikalakis, N. M., 2003. An algorithm for optimal free-form object matching. Computer-Aided Design 35(10), pp. 913–923.

Krivic, J. and Solina, F., 2004. Part-level object recognition using superquadrics. CVIU 95(1), pp. 105–126.

Laser scanner survey, 2005. Laser Scanner Survey. `www.pobonline.com/POB/FILES/HTML/PDF/POB0105-LaserScanningSurvey.pdf`.

Leica HDS, 2005. Leica Geosystems HDS. `http://hds.leica-geosystems.com/`.

Leica TPS, 2005. Leica Geosystems Total Stations. `http://www.leica-geosystems.com/corporate/en/products/total_stations/lgs_4207.htm`.

Leonardis, A., Jaklic, A. and Solina, F., 1997. Superquadrics for segmenting and modeling range data. PAMI 19(11), pp. 1289–1295.

Lichti, D. D., 2004. A resolution measure for terrestrial laser scanners. In: IAPRS, ISPRS XX Congress, Vol. XXXV, B3, Istanbul, Turkey, p. 6.

Lindström, P., 1983. A general purpose algorithm for nonlinear least squares problems with nonlinear constraints. Technical Report UMINF–102.83, Institute of Information Processing, University of Umeå, Sweden.

Lindström, P., 1984. Two user guides, one (ENLSIP) for constrained — one (ELSUNC) for unconstrained nonlinear least squares problems. Tech. Report UMINF–109.82 and 110.84, Institute of Information Processing, University of Umeå, Sweden.

Lindström, P. and Wedin, P.-Å., 1986. Methods and software for nonlinear least squares problems. Tech. Report UMINF–133.87, Institute of Information Processing, University of Umeå, Sweden.

Lowe, D. G., 1991. Fitting parameterized three-dimensional models to images. PAMI 13(5), pp. 441–450.

Lucchese, L., Doretto, G. and Cortelazzo, G. M., 2002. A frequency domain technique for range data registration. PAMI 24(11), pp. 1468–1484.

Luenberger, D. G., 1984. Linear and Nonlinear Programming. Addison-Wesley, Massachusetts.

Lukács, G., Martin, R. and Marshall, D., 1998. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In: ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume I, Springer-Verlag, pp. 671–686.

Lutton, E., Maitre, H. and Lopez-Krahe, J., 1994. Contribution to the determination of vanishing points using hough transform. PAMI 16(4), pp. 430–438.

Ma, W. and Kruth, J.-P., 1995. Parameterization of randomly measured points for least squares fitting of b-spline curves and surfaces. Computer-Aided Design 27(9), pp. 663–675.

Marshall, A. D., Lukács, G. and Martin, R. R., 2001. Robust segmentation of primitives from range data in the presence of geometric degeneracy. PAMI 23(3), pp. 304–314.

Masuda, T., 2002. Registration and integration of multiple range images by matching signed distance fields for object shape modeling. CVIU 87(1-3), pp. 51–65.

Maver, J. and Bajcsy, R., 1993. Occlusions as a guide for planning the next view. PAMI 15(5), pp. 417–433.

Mensi, 2002a. Decontamination of a Nuclear plant for Trnava Inc. Slovakia. `http://www.mensi.com/Website2002/cases/Vuje030618.pdf`.

Mensi, 2002b. Mensi's 3D Laser Scanning System Reduces Costs for Shell. `http://www.mensi.com/Website2002/cases/shell_02.pdf`.

Mensi, 2005. Mensi GS200. `http://www.mensi.com/Website2002/gs200.asp`.

Min, J., Powell, M. W. and Bowyer, K. W., 2000. Progress in automated evaluation of curved surface range image segmentation. In: ICPR, pp. 1644–1647.

Morita, S., 1999. Scale space range image aspect graph. In: SCALE-SPACE '99: Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision, Springer-Verlag, London, UK, pp. 465–470.

Mortenson, M. E., 1985. Geometric modeling. John Wiley & Sons, Inc., New York, NY, USA.

Natonek, E., 1998. Fast range image segmentation for servicing robots. In: The IEEE International Conference on Robotics and. Automation, pp. 406–411.

Nourse, B. E., IIakala, D. G., Hillyard, R. C., and Malraison, P. J., 1980. Natural quadrics in mechanical design. In: Proceedings of Autofact West 1, Anaheim, CA., pp. 363–378.

Olson, C. F., 2001a. A general method for geometric feature matching and model extraction. IJCV 45(1), pp. 39–54.

Olson, C. F., 2001b. Locating geometric primitives by pruning the parameter space. Pattern Recognition 34(6), pp. 1247–1256.

O'Rourke, J., 1998. Computational Geometry in C. Cambridge University Press, New York, NY, USA.

Parvin, B. and Medioni, G., 1986. Segmentation of range images into planar surfaces by split and merge. In: CVPR '86, pp. 415–417.

Petitjean, S., 2002. A survey of methods for recovering quadrics in triangle meshes. ACM Computing Surveys 34(2), pp. 211–262.

Pot, J., Thibault, G. and Levesque, P., 1997. Techniques for cad reconstruction of 'as-built' environments and application to preparing for dismantling of plants. Nuclear Engineering and Design 178(1), pp. 135–143.

Pottmann, H., Leopoldseder, S. and Hofer, M., 2002. Approximation with active b-spline curves and surfaces. In: PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, IEEE Computer Society, Washington, DC, USA, p. 8.

Pottmann, H., Leopoldseder, S. and Hofer, M., 2004. Registration without icp. CVIU 95(1), pp. 54–71.

Powell, M. W., Bowyer, K. W., Jiang, X. and Bunke, H., 1998. Comparing curved-surface range image segmenters. In: Proceedings of the Sixth ICCV, IEEE Computer Society, Washington, DC, USA, p. 286.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T., 1988. Numerical recipes in C: the art of scientific computing. Cambridge University Press.

Rabbani, T. and Heuvel, F. A. v. d., 2004a. 3d industrial reconstruction by fitting csg models to a combination of images and point clouds. In: IAPRS, Vol. XXXV, B3, Istanbul, Turkey, pp. 7–12.

Rabbani, T. and Heuvel, F. A. v. d., 2004b. Methods for fitting csg models to point clouds and their comparison. In: The 7th IASTED International Conference on Computer Graphics and Imaging, Kauai, Hawaii, USA, pp. 279–284.

Ramapriyan, H. K., 1976. A multilevel approach to sequential detection of pictorial features. IEEE Transactions on Computers 25(1), pp. 66–78.

Requicha, A. A. G. and Voelcker, H. B., 1982. Solid modeling: a historical summary and contemporary assessment. IEEE Computer Graphics and Applications 2(2), pp. 9–24.

Riegl, 2005. Riegl 3D Imaging Sensor LMS-Z360i. `http://www.riegl.com/terrestrial_scanners/lms-z360i_/360i_all.htm`.

Rosenfeld, A., 1969. Picture processing by computer. ACM Computing Surveys 1(3), pp. 147–176.

Rusin, D., 2004. Topics on Sphere Distribution. `http://www.math.niu.edu/~rusin/known-math/95/sphere.faq`.

Rusinkiewicz, S. and Levoy, M., 2001. Efficient variants of the ICP algorithm. In: Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling, pp. 145–152.

Russell, S. J. and Norvig, P., 2003. Artificial Intelligence: A Modern Approach. Pearson Education.

Sanders, F. H., 2001. 3d laser scanning helps chevron revamp platform. Oil-and-Gas-Journal.

Sanins, J., 2004. Capturing today for reuse tomorrow: Aveva's strategy for profitable as-built engineering. White paper, AVEVA Solutions Limited, Cambridge UK.

Sappa, A. D. and Devy, M., 2001. Fast range image segmentation by an edge detection strategy. In: Third International Conference on 3-D Digital Imaging and Modeling, pp. 292–299.

Schroeder, W., Martin, K. M. and Lorensen, W. E., 1998. The visualization toolkit (2nd ed.): an object-oriented approach to 3D graphics. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Schuhmacher, S. and Böhm, J., 2005. Georeferencing of terrestrial laser scanner data for applications in architectural modeling. In: S. El-Hakim, F. Remondino and L. Gonzo (eds), Proceedings of the ISPRS 3D-ARCH 2005: Virtual Reconstruction and Visualization of Complex Architectures, Vol. XXXVI, ISPRS, Mestre-Venice, Italy.

Schutz, C., Jost, T. and Hugli, H., 1998. Multi-feature matching algorithm for free-form 3d surface registration. In: Proceedings of the Fourteenth International Conference on Pattern Recognition, Vol. 2, pp. 982–984.

Scott, W. R., Roth, G. and Rivest, J.-F., 2003. View planning for automated three-dimensional object reconstruction and inspection. ACM Computing Surveys 35(1), pp. 64–96.

Sequeira, V., Ng, K., Wolfart, E., Goncalves, J. and Hogg, D., 1999. Automated reconstruction of 3d models from real environments. ISPRS Journal of Photogrammetry and Remote Sensing 54(1), pp. 1–22.

Shi, J. and Malik, J., 1997. Normalized cuts and image segmentation. In: CVPR '97, IEEE Computer Society, Washington, DC, USA, p. 731.

Shoemake, K., 1985. Animating rotation with quaternion curves. In: SIGGRAPH 1985, Vol. 19, ACM Press, pp. 245–254.

Sithole, G. and Vosselman, G., 2003. Automatic structure detection in a point-cloud of an urban landscape. In: 2nd Joint Workshop on Remote Sensing and Data Fusion over Urban Aereas (Urban 2003).

Skocaj, D. and Leonardis, A., 2001. Robust recognition and pose determination of 3-d objects using range images in eigenspace approach. In: Third International Conference on 3-D Digital Imaging and Modeling, IEEE Computer Society, pp. 171–178.

Söderkvist, I., 1999. Introductory overview of surface reconstruction methods. Technical Report 1999-10, Department of Mathematics, Lulea University of Technology, Lulea, Sweden. Can be retrieved from http://www.sm.luth.se/ inge/publications/surfrec.ps.

Spatial, 2004. The 3D ACIS ®Modeler (ACIS). `http://www.spatial.com`.

Srinivas, M. and Patnaik, L. M., 1994. Genetic algorithms: A survey. IEEE Computer 27(6), pp. 17–26.

Stamos, I. and Allen, P. K., 2002. Geometry and texture recovery of scenes of large scale. CVIU 88(2), pp. 94–118.

Stamos, I. and Leordeanu, M., 2003. Automated feature-based range registration of urban scenes of large scale. In: CVPR, pp. 555–561.

STAR, 2004. STAR–Services and Training through Augmented Reality. `http://www.realviz.com/STAR/objectives.htm`.

Stewart, C. V., 1999. Robust parameter estimation in computer vision. SIAM Review 41(3), pp. 513–537.

Strat, T. M. and Fischler, M. A., 1991. Context-based vision: Recognizing objects using information from both 2d and 3d imagery. PAMI 13(10), pp. 1050–1065.

Tang, C.-K. and Medioni, G., 1999. Robust estimation of curvature information from noisy 3d data for shape description. In: Proceedings of the Seventh International Conference on Computer Vision, Kerkyra, Greece, pp. 426–433.

Tangelder, J. W. H., Ermes, P., Vosselman, G. and Heuvel, F. A. v. d., 1999. Fitting parameterised object models to gradient images. In: Proceedings of the 5th annual conference of ASCI, pp. 449–456.

Tangelder, J. W. H., Ermes, P., Vosselman, G. and Heuvel, F. A. v. d., 2003. Cad-based photogrammetry for reverse engineering of industrial installation. Computer-Aided Civil and Infrastructure Engineering 18(1), pp. 264–274.

Taubin, G., 1993. An improved algorithm for algebraic curve and surface fitting. In: Third ICCV, IEEE, Berlin, pp. 658–665.

Trucco, E. and Fisher, R. B., 1995. Experiments in curvature-based segmentation of range data. PAMI 17(2), pp. 177–182.

Tucker, T. M. and Kurfess, T. R., 2003. Newton methods for parametric surface registration. part-i. theory. Computer-Aided Design 35(1), pp. 107–114.

Várady, T., Martin, R. R. and Cox, J., 1997. Reverse engineering of geometric models - an introduction. Computer-Aided Design 29(4), pp. 255–268.

Vosselman, G., 1995. Applications of tree search methods in digital photogrammetry. ISPRS Journal of Photogrammetry and Remote Sensing 50(4), pp. 29–37.

Vosselman, G., 1999. Building reconstruction using planar faces in very high density height data. In: IAPRS, Vol. 32, Munich, Germany, pp. 87–92.

Vosselman, G. and Dijkman, S., 2001. 3d building model reconstruction from point clouds and ground plans. In: IAPRS, Vol. 34, Annapolis, MA, USA, pp. 37– 44.

Vosselman, G., Gorte, B. G. H., Sithole, G. and Rabbani, T., 2004. Recognising structure in laser scanner point clouds. In: International Conference on Laser-Scanners for Forest and Landscape Assessment and Instruments, Processing Methods and Applications, Freiburg, Germany, pp. 33–38.

Wang, W., Pottmann, H. and Liu, Y., 2004. Fitting b-spline curves to point clouds by squared distance minimization. Technical Report TR-2004-11, Department of Computer Science, The University of Hong Kong, Hong Kong SAR, China.

Wani, M. A. and Arabnia, H. R., 2003. Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network. Journal of Supercomputing 25(1), pp. 43–62.

Wani, M. A. and Batchelor, B. G., 1994. Edge-region-based segmentation of range images. PAMI 16(3), pp. 314–319.

Webb, A., 1999. Statistical Pattern Recognition. Defence Evaluation and Research Agency, UK.

Weisstein, E. W., 2005. ”Kurtosis.” From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/Kurtosis.html`.

Werghi, N., Ashbrook, A., Fisher, R. B. and Robertson, C., 1999a. Faithful recovering of quadric surfaces from 3d range data. In: 2nd International Conference on 3D Digital Imaging and Modeling, Ottawa, Canada, pp. 280–289.

Werghi, N., Fisher, R. B., Ashbrook, A. and Robertson, C., 1999b. Shape reconstruction incorporating multiple nonlinear geometric constraints. Computer Aided Design 31(6), pp. 363–399.

Willard, D. E., 1985. New data structures for orthogonal range queries. SIAM Journal of Computing 14(1), pp. 232–253.

Wong, K. W., 1980. Basic mathematics of photogrammetry, section 2.5. In: C. Slama (ed.), Manual of Photogrammetry, 4th Edition, ASPRS, pp. 37–101.

Xiang, R. and Wang, R., 2004. Range image segmentation based on split-merge clustering. In: 17th ICPR, pp. 614–617.

Yu, Y., Ferencz, A. and Malik, J., 2001. Extracting objects from range and radiance images. IEEE Transactions on Visualization and Computer Graphics 7(4), pp. 351–364.

Zhang, Y., 2003. Experimental comparison of superquadric fitting objective functions. Pattern Recognition Letters 24(14), pp. 2185–2193.

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. IJCV 13(2), pp. 119–152.

Zhang, Z., 1997. Parameter estimation techniques: a tutorial with application to conic fitting. Image Vision Computing 15(1), pp. 59–76.

Zhao, D. and Zhang, X., 1997. Range-data-based object surface segmentation via edges and critical points. IEEE Transactions on Image Processing 6(6), pp. 826–830.

| | |
|---|---|
| ACM | Association for Computing Machinery |
| ASCI | Advanced School for Computing and Imaging |
| ASPRS | American Society for Photogrammetry and Remote Sensing |
| BMVC | Brithsh Machine Vision Conference |
| CIPA | The International Committee for Architectural Photogrammetry |
| CVGIP | Computer Vision, Graphics, and Image Processing |
| CVIU | Computer Vision and Image Understanding |
| CVPR | Conference on Computer Vision and Pattern Recognition |
| DAGM | DAGM (German Association for Pattern Recognition |
| ECCV | European Conference on Computer Vision |
| IAPRS | International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences |
| IASTED | International Association of Science and Technology for Development |
| ICCV | International Conference on Computer Vision |
| ICPR | International conference of Pattern Recognition |
| ICVS | International Conference on Computer Vision Systems |
| IEEE | Institute of Electrical and Electronics Engineers |
| IJCV | International Journal of Computer Vision |
| ISPRS | International Society for Photogrammetry and Remote Sensing |
| IVC | Image and Vision Computing |
| JOSA | Journal of the Optical Society of America |
| JPRS | ISPRS Journal of Photogrammetry and Remote Sensing |
| PAMI | IEEE Transactions on Pattern Analysis and Machine Intelligence |
| SIAM | Society for Industrial and Applied Mathematics |
| SIGGRAPH | Special Interest Group on Graphics and Interactive Techniques |
| SPIE | The International Society for Optical Engineering |

# Curriculum Vitae

Tahir Rabbani was born in 1976 in Chakwal, Pakistan. After completing his higher secondary education at Cadet College Hassan Abdal, he started his studies at the Department of Electrical Engineering at the University of Engineering and Technology Lahore in 1995. He completed his B.E. in Electrical and Communications Engineering in 2000 with distinction. His final year thesis was on using ray tracing based methods for the prediction and performance evaluation of indoor wireless networks. He joined Delta Indus Systems (www.deltaindus.com) as a Machine Vision Engineer in March 2000. There he worked on developing machine vision systems and algorithms for 3D measurement and analysis of surfaces using structured light. The developed systems were primarily used for 3D inspection of solder paste by SMT (Surface Mount Technology) production lines. Later, he worked as the Project Manager for designing a system for Automated Optical Inspection (AOI) of printed circuit boards.

He joined the Section of Photogrammetry and Remote Sensing of TU Delft as a PhD student in April 2002. Till June 2004, he worked there as a part of an EU project named STAR (Services and Training through Augmented Reality). The focus of the conducted research was to investigate various ways to automate the task of reconstructing 3D CAD models of industrial sites using point clouds and images. After the completion of the project he kept working on the same task whose results are reported in this thesis. Currently he is working as an R&D Engineer for Automated Test Equipment section of Agilent Technologies in Dublin, Ireland.