3D Object Recognition Based on Monocular Vision

Elizabeth González García
Departamento de Ingeniería Eléctrica, Electrónica, Automática y Comunicaciones
UNIVERSIDAD DE CASTILLA-LA MANCHA

Supervisors
Vicente Feliú Battle
Antonio Adán Oliver

A thesis submitted for the degree of
Doctor of Philosophy in Mechatronic

2012 April
Abstract

3D object recognition is one of the ultimate goals in computer vision and has been in the field’s focus since its early beginnings. Despite continuous efforts over several years, applications such as object manipulation remain unsolved and are subject to active research.

This thesis improves the state-of-the-art of 3D object recognition with monocular cameras. A thorough analysis of 2D shape representation models and similarity measures is developed. Shape representation models (region-based and contour-based) are combined with several similarity measures (deterministic and stochastic models) and tested using synthetic object models. These tests are focused on evaluating the performance of 2D shape recognition systems to recognize 3D objects.

A multiview framework with which to enhance the 3D object recognition based on one view is developed. Our framework is particularly addressed towards dealing with the uncertainty reduction problem using the active recognition paradigm. A mathematical background is presented in order to implement different non stochastic methodologies, with the goal of reducing uncertainty. The D-Sphere framework has several advantages over the other frameworks shown in the references, of which the flexibility in the design of the shape recognition and the active strategy, and the capability to reduce the uncertainty and the robustness to errors during the hypothesis estimation are particularly notable. An in-depth comparative analysis of different models with which to implement the proposed framework is also included.

Furthermore, we propose a dataset representation model that will allow object recognition to be developed in large multi-view object datasets. We present the Canonical Sphere Section (CSS) as a solution that highly reduces a view-based object database by considering the non-redundant information of the objects. We also define the NSSD (Normalized Silhouette in the Fourier Spectral Dominium)
descriptors to achieve a normalized shape representation of the object which is invariant to affine and geometrical transformations. The CSS-NSSD model has been experimentally tested and compared with other proposals. Based on the dataset representation model, an active recognition system is addressed in order to simplify the D-Sphere structure and thus make the recognition active method simpler and faster. Most of the ambiguity problems result from symmetries in the object model. Using all the nodes of the tessellated sphere therefore implies storing redundant views and information in the representation model. The proposed object recognition system is focused on satisfying the main constraints of any recognition system: high computational efficiency, high recognition rate and high accuracy for object pose estimation. The good performance of the proposed object recognition system is based on an efficient and normalized object representation model.

All experiments throughout this thesis are performed on challenging real world data. The proposed framework and the dataset representation model have been integrated as a part of an intelligent robotic task concerning the development of object manipulation. The recognition system particularly helps the robot to carry out a reliable grasping activity.
La tarea de reconocimiento de objetos 3D es un tema ampliamente abordado por los investigadores en el campo de la visión por ordenador. A pesar de los esfuerzos por desarrollar métodos de reconocimiento de objetos aplicables a tareas robóticas tales como la manipulación de objetos, dichos métodos carecen de la robustez necesaria para ser aplicados.

En esta tesis se aborda el problema de reconocimiento de objetos empleando visión monocular. Con el objetivo de evaluar los sistemas de reconocimiento de objetos 3D basados en una sola vista (2D), se ha desarrollado un exhaustivo análisis cuantitativo/cualitativo del comportamiento de un conjunto de sistemas de reconocimiento de formas que emplean tanto modelos de representación de formas basados en el contorno o en la región así como medidas de similitud deterministas y estocásticas. Dados los problemas de incertidumbre presentes en los sistemas de reconocimiento, se propone un framework de reconocimiento de objetos basado en el paradigma de la visión activa. Este framework tiene como ventaja de ser flexible en cuanto al modelo de reconocimiento de formas, así como a la estrategia activa que se emplee para desarrollar el sistema activo de reconocimiento. Las pruebas experimentales demuestran su robustez en presencia de objetos ambiguos y su capacidad para reducir la incertidumbre empleando el mínimo número de movimientos del sensor.

Además, en esta tesis, se propone un nuevo modelo de representación de la información contenida en la base de datos (CSS-NSSD). Este modelo reduce considerablemente el número de vistas que modelan un objeto basándose en el estudio de las simetrías reflectivas de dicho objeto (CSS) y representa la apariencia del objeto en cada vista empleando un descriptor invariante a transformaciones afines denominado como NSSD. Basado en el framework desarrollado y en el modelo de representación CSS-NSSD, se propone un sistema activo de reconocimiento que tiene como cualidades la de presentar una alta eficiencia computacional, alta tasa de reconocimiento y gran precisión en la estimación de la pose del objeto. Dicho sistema de reconocimiento ha sido implementado para desarrollar una aplicación de manipulación de objetos empleando un robot Staüblí.
To my husband and my parents
Acknowledgements

First of all, I would like to thank to my supervisor, Prof. Vicente Feliú, who opened the door of my postgraduate studies and led me into a rich and colorful life in Spain at the University of Ciudad Real. His deep insights and the suggestions made during his guidance have played a decisive role in the selection of the goals for my future life, but he mainly extended his hand to me when I most needed it.

I would like to thank to my supervisor, Prof. Antonio Adán. Thanks to the many occasions on which enlightening and effective discussions took place with Prof. Adán, I was able to publish my first paper after all the reviews he carried out. I have learned a lot from his serious and industrious working spirit during this process. He is not only an enthusiastic, kind hearted and experienced advisor in research, but also a good teacher in life.

My labmates Jonathan Becedas and Ismael Payo were of great assistance to my research progress. I have beneficed from their ways of thinking and styles of working. To Pedro Hungría for making my crazy ideas come true, and for his constructive observations. To Raúl González for his inestimable help. To the Image Processing staff from the UTIA Institute in Prague for share ideas and amazing times.

Besides, I can never forget coffee time with all my labmates Gabriela, Paco, Claudia, Choco, Gonzalo, Juan Carlos and Victor. We shared opinions about the future. Last but not the least, I want to thank to my friend Marcella, my parents and my husband, Pirra. They mean everything to me and they support me the most. I am refreshed whenever I think of them. I love you guys!
Contents

List of Figures vii
List of Tables xi

1 Introduction 1
1.1 Thesis framework ........................................... 3
1.2 Goals ....................................................... 3
1.3 Thesis outline ............................................... 5

2 View-Based 3D Object Recognition: An experimental comparative study 7
2.1 Introduction ................................................... 7
2.1.1 An Overview on Object Recognition ....................... 7
2.1.2 Main Problems in OR .................................... 9
2.1.3 3D Object Recognition with Monocular Vision ........... 11
2.2 2D Shape Recognition for 3D Environments. State of the art ..... 13
2.2.1 2D Shape Representation .................................. 13
2.2.2 Similarity Measures ....................................... 15
2.2.3 Shape Pose Estimation ................................... 16
2.2.4 Motivation of a Comparative Study ........................ 17
2.3 Statement of the Experimental Tests .......................... 18
2.3.1 Recognition Methods Set (RMS) .......................... 19
2.3.1.1 Computing shape descriptors ....................... 20
2.3.2 Experimental Platforms .................................. 21
2.3.3 Evaluation Parameters ................................... 22
2.4 Recognition Using Platform 1 ................................ 24
2.4.1 Platform Setup ............................................. 24
CONTENTS

2.4.2 Analysis of $Hr, Wr$ and $Ar$ Parameters ........................................ 24
2.4.3 Computational Costs ................................................................. 30
2.4.4 Pose Estimation Accuracy ............................................................. 31
2.4.5 Object Identification Rates ........................................................... 33
2.5 Recognition Tests Using Platform 2 ..................................................... 34
  2.5.1 Platform Setup ............................................................................. 34
  2.5.2 Analysis of $Hr, Wr$ and $Ar$ parameters ........................................ 37
  2.5.3 Sensitivity to the number of views of the 3D representation model ....... 40
  2.5.4 Sensitivity to the number of features ............................................. 42
  2.5.5 Comparison Results for Different Object Datasets .......................... 43
2.6 Results Discussion and Conclusions ...................................................... 47

3 Multiple View-Based 3D Object Recognition. A framework for active recognition systems ......................................................... 51
  3.1 Introduction ...................................................................................... 51
  3.2 Previous works in Active Strategies in 3D Object Recognition Problems .... 53
  3.3 Active Recognition Framework ......................................................... 56
    3.3.1 Shape representation ................................................................. 56
    3.3.2 Shape recognition ................................................................. 58
    3.3.3 Sensor displacement model ....................................................... 59
    3.3.4 Active recognition ................................................................. 62
      3.3.4.1 Task 1: Obtaining new view candidates ................................... 64
      3.3.4.2 Task 2: Determination of the decision model ......................... 66
      3.3.4.3 Task 3: Next best view decision ............................................. 68
  3.4 Active Strategies Methodologies ......................................................... 71
    3.4.1 Possibilistic method ($\Pi$) .......................................................... 71
    3.4.2 Heuristic method ($H$) ............................................................. 72
    3.4.3 Deterministic method ($\Delta$) ..................................................... 73
  3.5 Experimentation ............................................................................... 73
    3.5.1 General statement of the experimental tests ................................. 74
    3.5.2 Experiments 1 ........................................................................... 75
    3.5.3 Experimentation 2 ................................................................... 82
      3.5.3.1 Robotic Platform ............................................................... 83
4 Active Object Recognition based on CSS-NSSD

4.1 Introduction .................................... 89

4.2 A simplified Representation Model .................. 91
  4.2.1 Defining the Canonical Spherical Section (CSS) .... 91
  4.2.2 Defining the Canonical Spherical Section (CSS) .... 92
  4.2.3 Computing the Canonical Spherical Section (CSS) .... 93

4.3 NSSD descriptor .................................. 96
  4.3.1 Affine Normalization ........................... 97
  4.3.2 Precision Normalization and Dimensionality Reduction .... 99

4.4 NSSD descriptor properties ............................ 99

4.5 Estimation of pose parameters ........................ 102

4.6 Using the Active Recognition Framework .............. 104
  4.6.1 CSS-NSSD System. Implementation Details ........ 104

4.7 Experimentation .................................. 107
  4.7.1 Experiment 1 .................................. 108
  4.7.2 Experiment 2 .................................. 109
    4.7.2.1 Recognition and computational evaluation on a robotic platform ..................... 110
    4.7.2.2 Pose error evaluation in a simulator .............. 112

4.8 Final Discussions and Conclusions ........................ 113

5 A Grasping Experience with the CSS-NSSD Approach 115

5.1 Introduction ..................................... 115

5.2 Visual Control ................................... 116
  5.2.1 Open loop Robot Control or Look-And-Move ............ 116
  5.2.2 Closed Loop Robot Control (Visual Servoing) ........... 118

5.3 Grasping Planner .................................. 119
  5.3.1 An Overview .................................. 119
  5.3.2 Grasping synthesis: contact model and grasp-points coordinates ...................... 121
  5.3.3 Kinematics of the robot ........................ 123
  5.3.4 Path planning ................................. 129
CONTENTS

5.4 Experimental test .............................................. 130

6 Conclusions and Future Works 137

6.1 Contributions .................................................. 137
6.2 Contrast of Results ........................................... 139
6.3 Future Improvements and Developments .................... 140

Bibliography ....................................................... 143
List of Figures

2.1 General scheme for a shape recognition system ........................................... 18
2.2 Object example from ALOI-VIEW collection viewed from 72 different viewpoints (figure from [45]) ................................................................. 25
2.3 Objects used in the experimental tests. ............................................................ 25
2.4 The training set for one object and view is: a) prototype image and two views close to the ground truth, b) six images obtained after applying affine transformations to the prototype and c) six images with different Gaussian noise levels injected into the prototype image. ................................................................. 26
2.5 Noise effects on the binary images after the image preprocessing ................. 26
2.6 Processing time (in seconds) needed to obtain the representation model. ...... 31
2.7 Processing time needed for similarity measures. Plots from a) and b) show the time rates according to the number of descriptors in the shape representation model. c) Times rates according to the number of views in the dataset for shape models using 128 descriptors. ................................................................. 32
2.8 Object identification rates ($I_r$). ................................................................. 34
2.9 Samples of 3DSL Synthetic collection. ......................................................... 35
2.10 Image depth extraction. a) 3D object and the arrow corresponding to a selected viewpoint. b) Object model viewed from the selected viewpoint. c) Image depth. 35
2.11 Training images from a node of the tessellated sphere. The first row shows images around a sphere node, the second row shows the image with Gaussian, salt and pepper and sparkle noise. ......................................................... 36
LIST OF FIGURES

2.12 Experimental setup. The experimental platform uses a Stabli robot with a we-
bcam on the end-effector. The robotic vision system captures images around
the object. The robot positions correspond to nodes of an imaginary tessellate
sphere centred in the scene. .................................................. 36
2.13 Images corresponding to various tests. ........................................ 37
2.14 a) Recognition mean relative variation for meshes of 80 and 320 nodes for dif-
ferent similarity measures. b) Recognition mean relative variation for different
2D shape representation models. ............................................ 43
2.15 Tendency of parameter $H_r$ versus the relative number of features with regards
to the canonical number. .................................................. 44
2.16 a) Recognition mean relative variation for meshes of 80 and 320 nodes for dif-
ferent similarity measures. b) Recognition mean relative variation for different
2D shape representation models. ............................................ 47

3.1 General scheme of an active recognition system based on monocular vision. ... 52
3.2 Robotic Platform Configuration. ........................................... 57
3.3 New view candidates obtaining process. A set $B^{(k)}$ of view candidates (or hy-
oposes) is determined for an image $I^{(k)}$ captured at step $k$ and the accumulated
evidence $E^{(k)}_l$ and accumulated Euler representation vectors $T^{(k)}_l$ are updated for
each of the candidate views $b_l \in B^{(k)}$. .................................................. 66
3.4 Process used to build the D-Sphere and the next best view selection. a) Show
for the best view candidate $b_\ast$, the sensor positions that correspond to pointing
towards the centre of the tessellated sphere through each node $m$ at a distance
$d$ if $b_\ast$ is verified. b) D-Sphere estimation. The level of ambiguity among the
views mapped in the node is computed for each D-Sphere node. c) Next best
view selection based on a decision function $J(\rho_m, g_m, \varsigma_m)$ ............... 69
3.5 Example of the proposed framework performance in case uncertainty by ambi-
guity. .............................................................................. 70
3.6 Experimental setup. a) The experimental platform uses a Staübl robot with a
camera on the end-effector. b) Synthetic objects in Dataset 1. c) Object model
placed in the tessellated sphere and some views. d) Samples of images captured
in the vision-robot setup and used during the active recognition tests. ........... 75
3.7 Possibilistic framework membership functions. a) Membership function for $X_{fi}$; b) Membership function for $D(\vec{w}_i, \bar{w}(k))$; c) Output membership function. 76

3.8 Recognition Rate in each step using different shape recognition methods. Each plot shows the active recognition rate for the selected descriptor with different active strategies (Posibilistic ($\Pi$), Heuristic($H$), Deterministic($\Delta$)). a) Fourier Descriptors (FD). b) Boundary Moments (HM). c) Principals Components (PCA). d) Zernike Moments (ZM). e) Complex Moments (CM). 78

3.9 Recognition Rate in each step using different active strategy methods. Each plot shows the active recognition rate for the selected active strategy with different shape descriptors (FD, HM, PCA, ZM, CM). a) Posibilistic ($\Pi$). b) Heuristic($H$). c) Deterministic($\Delta$) 79

3.10 Recognition costs for the active recognition strategies. a) Time rates for a single position of the camera taking different combinations between active strategies and shape representation model ($\Sigma_c$). b) Time rates for the different active strategies when the number of hypotheses increases. 80

3.11 Active Recognition Performance. a) Recognition cost ($C$). b) Recognition Rate ($R$) 81

3.12 Computational efficiency ($C(s)$) considering $g_m = 1$ 81

3.13 Active recognition systems performance for Dataset 1, Dataset 2 and Dataset 3. a) Recognition Rate ($R$)(%) b) Recognition cost ($C(s)$). 84

3.14 Examples of pose estimation using different active recognition systems. (a) AR 1. (b) AR 2. (c) AR 3. 86

4.1 a) Sphere coordinates system b) Object principal axes. c) Object principal axis aligned with the reference coordinate system. 92

4.2 Samples of objects with different symmetries, their Canonical Object Section and their Canonical Sphere Section.(a) $n_y = 1$ and $n_z = 0$. (b) $n_y = 2$ and $n_z = 0$. (c) $n_y = 4$ and $n_z = 0$. (d) $n_y = 20$ and $n_z = 2$. 92

4.3 Example of the relationship between one view (image) from the Canonical Spherical Section and its corresponding views in the in the canonical sphere. 96

4.4 Example of an object Canonical Spherical Section (CSS) model. 97
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>Example of NSSD descriptor invariance property. (a) Samples of affine transformations for the original Silhouette. (b) Superposition of NSSD descriptors (module) for different affine transformations.</td>
</tr>
<tr>
<td>4.6</td>
<td>Example of stability property. (a) The original silhouette. (b) Distorted silhouette. (c) NSSD descriptor module for the original and the distorted silhouette. The mean square error is 0.0014852. (d) FD descriptor module for the original and the distorted silhouette. The mean square error is 0.0030151</td>
</tr>
<tr>
<td>4.7</td>
<td>Example of completeness. (a) The original silhouette and distorted silhouette after applying an affine transformation. (b) Reconstructed silhouette from invariants NSSD.</td>
</tr>
<tr>
<td>4.8</td>
<td>Original silhouettes (red) and recovered silhouettes using the second, third and penultimate elements of NSSD descriptor vector (blue).</td>
</tr>
<tr>
<td>4.9</td>
<td>Experimental setup. a) The experimental platform uses a Staübli robot with a camera on the end-effector. b) Synthetic objects in the dataset. c) Samples of images captured in the vision-robot setup and used during the active recognition tests.</td>
</tr>
<tr>
<td>4.10</td>
<td>NSSD descriptor performance. (a) Recognition rates. (b) Shape pose error. (c) Computational cost.</td>
</tr>
<tr>
<td>5.1</td>
<td>Grasping planner outline.</td>
</tr>
<tr>
<td>5.2</td>
<td>a) Soft-finger contact b) Pairs of optimal contact points obtained by following the Adan’s algorithm.</td>
</tr>
<tr>
<td>5.3</td>
<td>Frames of PUMA robot.</td>
</tr>
<tr>
<td>5.4</td>
<td>Defining the robot tip frame at grasping position.</td>
</tr>
<tr>
<td>5.5</td>
<td>Detail of the two-finger gripper and the camera integrated in the end-effector.</td>
</tr>
<tr>
<td>5.6</td>
<td>Photograph of the experimental setup.</td>
</tr>
<tr>
<td>5.7</td>
<td>Sequence of robot movements during an object manipulation task. The manipulation system us based on AR 1 active recognition system based on the proposed framework.</td>
</tr>
</tbody>
</table>
List of Tables

2.1 2D shape representation methods and their characteristics. ................................. 14
2.2 Properties of several similarity measures. ................................................................. 16
2.3 Examples of test images. ......................................................................................... 27
2.4 Comparative analysis of combinations of different shape descriptors and various similarity measures using ALOI dataset (Hard Recognition (Hr), Weak Recognition (Wr) and Ambiguous Recognition (Ar)). ................................................................. 28
2.5 Average of the Quadratic Mean Error (Av(QME)) and execution time in pose estimation. ................................................................. 33
2.6 Comparative analysis of combinations of different shape descriptors and various similarity measures using 3DSL dataset (Hard Recognition (Hr), Weak Recognition (Wr) and Ambiguous Recognition (Ar)) ................................................................. 38
2.7 Comparative analysis of combinations of different shape descriptors and various similarity measures using 3DSL dataset (Hard Recognition (Hr), Weak Recognition (Wr) and Ambiguous Recognition (Ar)). The first row corresponds to Dataset with 80 views and the second row corresponds to 320 views (Dataset 3). ................................................................. 41
2.8 Comparative analysis of combinations of different shape descriptors and various similarity measures using two different 3DSL datasets. Dataset 1 includes dissimilar objects (objects 1, 3, 8, 14, 18), Dataset 2 has objects with some similarity (objects 2, 6, 7, 9, 12). ................................................................. 45
3.1 Collection of methods used to implement the proposed framework .......................... 76
3.2 Average of the Quadratic Mean Error (Error) and execution time (Time) in pose estimation ................................................................. 80
LIST OF TABLES

3.3 Collection of methods used to implement the proposed framework to develop a comparative analysis .............................................. 83
3.4 Comparison between different active recognition systems in the active simulator 85
4.1 Collection of methods used to implement the proposed framework to develop a comparative analysis ............................................. 106
4.2 Set of active recognition systems (ARS) and the main methods used by their main modules .............................................................. 111
4.3 Comparison between different active recognition systems on the robotic platform 111
4.4 Comparison between different active recognition systems in simulation ........ 113
5.1 DH parameters .......................................................... 124
5.2 Objects pose ............................................................ 133
5.3 Summary of results ..................................................... 134
Symbol Table

Chapter 2

\( Ar \): ambiguous recognition rate
\( Ir \): shape identification rate
\( Hr \): hard recognition rate
\( M \): shape representation model
\( N \): total number of tests
\( ne \): the number of samples correctly identified
\( np \): the number of weak solutions
\( nw \): the number of wrong solutions owing to ambiguity
\( S \): shape sample
\( (x_q, y_q) \): \( S \) coordinates
\( Wr \): weak recognition rate
\( P \): evaluation parameter \( \{Hr, Wr, Ar\} \)
\( QME \): quadratic mean error
\( Q \): total of elements in the shape coordinates vector
\( \hat{Q}_{rm,i} \): mean of the recognition score
\( q \): index of an element in the shape coordinates
\( \hat{R}_{rm,i} \): mean of the recognition score
\( S \): similarity measure
\( \Delta \hat{R}(P,S) \): recognition mean relative variation for each parameter \( P \) and similarity measure \( S \)
\( \Delta \hat{Q}(P,M) \): recognition mean relative variation for each parameter \( P \) and shape representation model \( M \).

Chapter 3

\( \alpha \): roll Euler angle
\( A \): Score the ambiguity
\( \beta \): pitch Euler angle
\( C \): Computational efficiency
\( \Delta \): Deterministic methodology
\( \delta_l^{(k)} \): translation vector in the pose between synthetic database image \( l \) and real view taken at step \( k \) \( (\delta_l^{(k)} \in \mathbb{R}^{2 \times 1}) \)
GLOSSARY

$x_δ^{(k)}$: translation component of the pose of image $I^{(k)}$ in the direction of axis $x_{jl}$ corresponding to the $jl$ sphere node

$y_δ^{(k)}$: translation component of the pose of image $I^{(k)}$ in the direction of axis $y_{jl}$ corresponding to the $jl$ sphere node

$ε$: dissimilarity threshold

$η$: evidence threshold

$γ$: yaw Euler angle

$H$: Heuristic methodology

$X$: Compute the accumulated evidence

$κ$: translation vector of the Euler representation that expresses the relative position between the origins of two frames ($κ ∈ ℝ^{3×1}$)

$λ_{kl}^{(k)}$: scaling factor in the pose between synthetic database image $l$ and real view taken at step $k$

$μ()$: function to recover $l$ from the pair $i, j$

$Φ()$: invariant feature vector operator

$Π$: Possibilitic methodology

$ψ_j$: azimuth coordinate of the spherical representation

$Ψ()$: function that provides the robot joint variables needed to move the sensor to a position that points towards the center of the tessellated sphere from a given distance with a direction defined by a given node ($Ψ ∈ ℝ^{6×1}$)

$Ψ^{-1}()$: function that, for a given hypothesis, a given object pose obtained under the hypothesis of having viewed $b_j$ with a certain sensor position, and a new specified sensor position, yields the node of the tessellated sphere which contains the view of the object that is most likely to be seen by the sensor.

$R$: Recognition rate

$ρ_m$: uncertainty/ambiguity score at node $m$

$σ_j$: polar coordinate of the spherical representation

$Θ•$: vector of joint variables ($Θ ∈ ℝ^{6×1}$)

$Θ_{m,k}•$: joint coordinates with which to point the object from node $m$ under hypothesis $b_{*,k}$

$θ_i$: $i^{th}$ joint variable (angle)

$ϒ()$: function that yields the homogeneous transform corresponding to a given Euler representation of a frame ($ϒ ∈ ℝ^{4×4}$)
GLOSSARY

\( \Upsilon^{-1}(\cdot) \): function that yields the Euler representation corresponding to a given homogeneous transform \((\Upsilon^{-1} \in \mathbb{R}^{6 \times 1})\)

\( \phi^{(k)} \): rotation angle in the pose between synthetic database image \( l \) and real view taken at step \( k \)

\( \varsigma(\cdot) \): function that defines the reachability of the robot in the configuration space

\( \varsigma_m \): reachability of node \( m \)

\( \bar{a}_1, \bar{a}_2, \ldots \bar{a}_n \): generic column vectors

\( B \): view database, or candidate view set

\( b_l \): hypothesis synthetic view (element of the database)

\( D(\cdot) \): function that scores the uncertainty/ambiguity for a set of \( n \) vectors

\( D() \): dissimilarity measure

\( d \): distance between the sensor and the center of the object

\( d_{lj}^{(k)} \): distance from the sensor origin to the center of the tessellated sphere

\( E(\cdot) \): accumulated evidence in a database element

\( E(\cdot) \): evidence function between a feature vector obtained from a real image and a database element

\( E_l^{(k)} \): evidence of hypothesis \( l \) obtained from the measurement at step \( k \)

\( G(\cdot) \): function that calculates the motion effort needed to transport the sensor from a previous location to a location associated with a given node

\( g_m \): "motion effort" of transporting the sensor from a previous location to a location associated with the node \( m \)

\( h_{m,l} \): auxiliary database element variable, signifying the element that would be seen from sensor position \( \Theta_{m,*}^{(k)} \) if hypothesis \( l \) were verified

\( I\bullet \): synthetic or real image

\( i\bullet \): object index

\( J(\cdot) \): decision function

\( J'(\cdot) \): a factor of the decision function

\( J \): number of sphere nodes

\( j\bullet \): node index

\( k \): iteration step

\( L \): number of elements in the view database

\( l \): index of an element in the view database

\( \hat{l} \): auxiliary index of database elements
GLOSSARY

\( m \cdot \): auxiliary node index

\( N \): number of objects in dataset

\( O \): object dataset

\( o_i \): the i-th object in the dataset

\( P() \): function that computes the pose set \( p \) between a synthetic database image and a real view

\( p_i^{(k)} \): pose set between synthetic database image \( l \) and a real view taken at step \( k \)

\( r \): sphere radius

\( R_{R_b}^{R_t}() \): homogeneous transform between the robot tip frame (\( R_t \)) and the robot base frame (\( R_b \)) (\( R_{R_b}^{R_t} \in \mathbb{R}^{4\times4} \))

\( S_{R_b}^{R_t}() \): homogeneous transform from the frame of the center of the tessellated sphere (\( S_b \)) to the robot base frame (\( R_b \)) (\( S_{R_b}^{R_t} \in \mathbb{R}^{4\times4} \))

\( S_{R_t}^{R_b}() \): homogeneous transform from the robot tip frame (\( R_t \)) to the frame in the center of the tessellated sphere (\( S_b \)) (\( S_{R_t}^{R_b} \in \mathbb{R}^{4\times4} \))

\( S_{R_n}^{R_t}() \): homogeneous transform from the robot tip frame (\( R_t \)) to a sphere node frame (\( R_n \)) (\( S_{R_n}^{R_t} \in \mathbb{R}^{4\times4} \))

\( T \cdot \): Euler representation vector (\( T \in \mathbb{R}^{6\times1} \))

\( T^{(k)} \): homogeneous transform between the frame in the center of the tessellated sphere (\( S_b \)) and the robot base frame (\( R_b \)), which represents the estimated pose of the object associated with view \( l \) with regard to the robot base (\( T_l \in \mathbb{R}^{4\times4} \)), and which may, optionally, have as a superscript the number of iteration (\( k \)).

\( \bar{u}_j \): unity vector that expresses the direction associated with a node (\( \bar{u} \in \mathbb{R}^{3\times1} \))

\( \bar{v} \cdot \): vector of features
GLOSSARY

\{\vec{v}_{hm}\}: set of feature vectors of images taken by the sensor if it is placed at \(\Theta_{m,*}^{(k)}\) under the different hypotheses \(l\)

\(\vec{w}\): vector of invariant features

\(\vec{x}_j\): unity vector of the frame associated with node \(j\) expressed in the frame in the center of the tessellated sphere (\(\vec{x}_j \in \mathbb{R}^{3x1}\))

\(\vec{y}_j\): unity vector of the frame associated with node \(j\) expressed in the frame in the center of the tessellated sphere (\(\vec{y}_j \in \mathbb{R}^{3x1}\))

\(\vec{z}_j\): unity vector of the frame associated with node \(j\) which has the opposite direction to \(\vec{u}_j\) expressed in the frame in the center of the tessellated sphere (\(\vec{z}_j \in \mathbb{R}^{3x1}\))

\(\vec{z}_s\): the roll unity vector of the sensor frame (\(\vec{z}_s \in \mathbb{R}^{3x1}\))

Index \(\bullet\) may be: 1) empty (generic value), 2) subscript \(l\) (belonging to database element \(b_l\)), 3) subscript \(i, j\) (belonging to synthetic view of object \(i\) from node \(j\)), 4) superscript \((k)\) (associated with the \(k^{th}\) measurement or movement), 5) combination of subscript \(l\) and superscript \((k)\).

\(*\) : subscript that substitutes subscript \(l\) in all the variables calculated under the hypothesis of having the best synthetic view candidate (\(l = *\))

**Chapter 4**

\(B\bullet\): view database, or candidate view set

\(b_l\): hypothesis synthetic view (element of the database)

\(C\): computational cost

\(\{c_r\}\): a cluster

\(D\): function that scores the uncertainty/ambiguity for a set of \(n\) vectors

\(D()\): dissimilarity measure

\(\delta\): displacement of the starting point

\(E()\): evidence function between a feature vector obtained from a real image and a database element

\(H E_l^{(k)}\): evidence of hypothesis \(l\) obtained from the measurement at step \(k\)

\(\mathcal{F}\): the Fourier transform

\(f\): cost function

\(g()\): function that calculates the "motion effort" needed to transport the sensor from a previous location to a location associated with a given node

\(g_m\): motion effort of transporting the sensor from a previous location to a location associated with the node \(m\)
GLOSSARY

(\(\bar{\gamma}\)): translation factor
\(\mathcal{H}\): length of NSSD descriptor
\(\bar{h}\): index for a NSSD element
\(I_*\): synthetic or real image
\(i\).: object index
\(J\): number of sphere nodes
\(\mathcal{J}\)(): decision function
\(j\): node index
\(K'\): number of significant values in the precision normalization process
\(k_{\bar{z}}\): index used to compute the \(n_{\bar{z}}\): reflective symmetry order
\(k_{\bar{z}}\): index used to compute the \(n_{\bar{z}}\): reflective symmetry order
\(L\): number of elements in the view database
\(l\): index of an element in the view database
\(M\): length of \(s\) represented in the Fourier spectral dominium
\(\mu\)., \(\rho\)., \(\tau\).: related to the normalization process
\(N\): silhoutte points
\(n\) index of a silhoutte point
\(n_{\bar{x}}\): reflective symmetry order in plane XZ
\(n_{\bar{y}}\): reflective symmetry order in plane XY
\(o\): object model
\(\Omega\): cluster set
\(P_e\): pose accuracy
\(\psi_j\): azimuth coordinate of the spherical representation
(\(\phi\)): rotation angle between two silhouettes
(\(\bar{\rho}\)): scale vector \(R\): recognition rate
\(\bar{R}\): number of clusters.
\(r\): cluster index.
\(S_{4}^{(K'_{\bar{x}})}\): \(S_4\) normalized to precision
\(S_{4}^{(K'_{\bar{y}})}\): NSSD descriptor
\(S_4\): silhouette normalized to affine transformation
\(\mathcal{T}\).: Euler representation vector \((\mathcal{T} \in \mathbb{R}^{6x1})\)
\(S\): canonical sphere
\(S\): sphere
GLOSSARY

s: silhouette
\( \sigma_j \): elevation coordinate of the spherical representation
\( \vec{v} \): vector of features
\( \hat{w} \): NSSD descriptor
\( \hat{w} \), the second, third and penultimate elements of \( \vec{w}_l \)
\( X_l \) representing the level of ambiguity
\( X(m) \): representation spectral of \( x(n) \) coordenate
\( X,Y,Z \): object canonical axes
\( \Xi^* \): canonical sphere section
\( \Xi \): the rest of the canonical sphere
\( (x(n),y(n)) \): cartesian coordinates
\( Y(m) \): representation spectral of \( y(n) \) coordenate

Index \( \bullet \) may be: 1) empty (generic value), 2) subscript \( l \) (belonging to database element \( b_l \)), 3) subscript \( i,j \) (belonging to synthetic view of object \( i \) from node \( j \)), 4) superscript \( k \) (associated with the \( k^{th} \) measurement or movement), 5) combination of subscript \( l \) and superscript \( k \).

Chapter 5
\( \vec{a} \): unitary vector corresponding to \( Z_6 \)
DH kinematic parameters
\( \alpha_i - 1 \)
\( a_i - 1 \)
\( d \)
\( \theta \)
\( \vec{n} \): unitary vector corresponding to \( X_6 \)
\( P_1, P_2 \): contact points coordenates
\( Rb \): robot base frame
\( (Rb P_1, Rb P_2) \): contact points coordinates in the robot base frame
\( \vec{s} \): unitary vector corresponding to \( Y_6 \)
\( Sb \): tessellated frame
\( T_k \): homogeneous transform between the frame in the center of the tessellated frame \( (Sb) \) that surrounded the object and the robot base frame \( (Rb) \)

\( \bullet T \): transformation matrix
\( \Theta = (\theta_1, ..., \theta_6) \): robot joints
GLOSSARY

$t_i$: the initial time of the trajectory
$tf$: the final time of the trajectory

Components of transformation matrix

$n_•$: elements of first row
$s_•$: elements of second row
$a_•$: elements of third row
$p_•$: elements of four row
1

Introduction

Object recognition (OR) is an important topic in the field of computer vision. In general, we can state that an object is recognized when the system is able to identify that object in the scene. Object recognition is thus a broad concept which does not necessarily imply others significant tasks in the field of image processing and 3D data processing, such as segmentation, registration or positioning. Furthermore, OR can be tackled from bidimensional or tridimensional viewpoints. From a bidimensional point of view, OR identifies specific information contained in the 2D image of the scene. This information, which is generically called the ‘object’, is a part of the knowledge database of the system. In the 2D case, the image contains a distorted version of the object which can be modeled by an affine transformation. From a tridimensional point of view, the OR problem is the same but the object is now considered to be a solid and can therefore be viewed from many directions. The unitary problem solved for 2D consequently becomes a multiple problem. The subject matter contained in this document deals with the tridimensional object recognition problem.

In a tridimensional environment OR usually implies not only determining the object’s identity, but also calculating the pose of the object with regard to a standard reference system. This leads the recognition systems to be integrated into others sensorial or computational systems with the goal of accomplishing more and extended intelligent tasks. For example, several techniques which address the problem of finding the position and orientation of a known object have been proposed for fully-automated production systems.

Object recognition systems have many applications, from automatic manufacturing [1], product inspection [2, 3], counting and measuring [4] to medical surgery [5, 6, 7]. They are
1. INTRODUCTION

often found in tasks which are hard to achieve with manual labor and which demand high accuracy and quality.

It is well known that the strategies used to develop 3D recognition systems are closely related to the sensor type and the setup used. The most popular sensors are range finders, stereo vision systems and monocular systems. Range finders capture images, providing surface information or depth of the scene [8, 9, 10]. Many industrial tasks have been efficiently accomplished with range data for many years [11]. However, in addition to the costly nature of these devices, these sensors also suffer problems such as shadows, and an inability to sense highly reflective surfaces and some color. The main drawback of stereo vision techniques is that they have to match the points from the left and right images provided by both cameras in a precise manner [12, 13]. This correspondence problem is not always correctly solved and is strongly dependent on the texture of the images. Moreover, the lack of texture in the image leads to ambiguities in the correspondence process, thus generating erroneous matches and, consequently, erroneous 3D data.

One of the frameworks in which the 3D object recognition systems have been integrated corresponds to the robotic field, principally in object manipulation, inspection and robot navigation. A sophisticated vision technology provides saving, quality, reliability, safety and productivity in robot applications. As was previously mentioned, these vision systems have already been integrated into industrial applications (e.g. manufacture automation). However, in the service robotic field, most of them are mere prototypes. More particularly, one of the basic tasks to be developed by robots is that of object manipulation. In this case, the system needs to solve many problems to achieve the final goal, including, among others, object recognition, pose estimation, grasping and path planning. 3D object recognition again arises as a key stage of the total process which must be carried out with a high degree of reliability. In this framework, most of existing works are unable to provide high accuracy of the object pose estimation and, consequently, do not guarantee the stability constraints in the grasping process. In others, the recognition problem cannot be efficiently solved when the object database includes objects which may be ambiguous from different viewpoints. The current vision systems in robots with manipulation purposes therefore have clear limitations that are worth researching.

This thesis presents an in-depth study of the 3D recognition problem and its integration into a manipulator robot. The approach developed integrates a camera into the end-effector of a manipulator robot with the aim of carrying out a reliable identification and accuracy posi-
1.1 Thesis framework

This thesis has been framed in project PBI-06-0105 “Autonomous robot for tank inspection” (sponsored by the Community Board of Castilla La Mancha) and developed in the Robotics Lab of Castilla La Mancha University. The final goal of this project is to develop an autonomous robot for tank inspection with a navigational capability on the ceiling and walls of the tank. The inspection system must be able to avoid obstacles and provide enough information to manipulate various pieces in the tank.

In this multidisciplinary project, the research developed in this thesis contributes to the building of a vision system which will help the robots to make manipulation decisions in the tank. A further intention of this thesis is to develop a generic framework for active recognition systems in which not only tank pieces but any free form object can be used. Some incomplete previous results of this project were developed by Andrés Vázquez’s [14] and Jonathan Becedas [15]. The first dealt with path planning of the robot arm including obstacle circumstances, whereas the second was concerned with the implementation of a robotic gripper, which was built with flexible materials. In order to successfully pick up the object both works require precise knowledge about the object position and orientation in the scene. In this respect, this thesis will contribute towards providing a more reliable and accurate knowledge of the scene to be manipulated.

1.2 Goals

The principal goal of this thesis is to propose a vision framework that is integrated into a manipulator robot in which to implement 3D active recognition algorithms based on monocular vision and with which to facilitate reliable manipulation robotics tasks.

This general objective can be divided in the following partial goals:

1. Study the problem of 3D object recognition using only one image.

   (a) Carry out an in-depth qualitative and quantitative analysis of the performance of 2D shape recognition methods when they are used to solve 3D object recognition problems.
1. INTRODUCTION

(b) Combine several of the most important 2D shape descriptors (contour and regions) with a set of representative similarity measurements (deterministic and stochastic) and provide an exhaustive experimental comparison of 3D shape recognition techniques when only one view of the scene is taken.

2. Integrate a vision system in a robotic setup to reduce the ambiguity problem in monocular vision

(a) Formulate the recognition problem with multiple views by means of a vision-robot system.

(b) Define a complete framework which will allow the ambiguity problem to be tackled for any generic object recognition system based on active recognition paradigms. This framework must be independent of the type of shape descriptors and similarity measurements used.

(c) Develop a mathematical background to implement different non stochastic methodologies with the goal of reducing uncertainty in the scene.

3. Propose, within the vision-robot framework, a particular object database representation model together with an efficient active recognition system.

(a) Define an object model representation which is invariant to affine transformations and is able to represent all the shapes in the dataset with the same precision.

(b) Define the active recognition system.

(c) Carry out an experimental comparison with other active recognition strategies.

4. Test the performance of the proposed active recognition system in robotic manipulation tasks.

(a) Design the experimental setup. Integrate a gripper into the vision-robot system.

(b) Establish a particular grasping solution for the tests.

(c) Perform and evaluate experiences of grasping using the proposed active recognition system.
1.3 Thesis outline

Chapter 1 has been addressed towards motivating and setting the main objectives of the thesis. The remainder of the document has been organized as follows.

Chapter 2 is entitled “Evaluation of 2D Shape Recognition Models applied to 3D Object Recognition”, and a comparative analysis of 2D shape representation models and similarities measures is developed in this chapter. Shape representation models (region-based and contour-based) are combined with several similarity measures (deterministic and stochastic models) and tested using synthetic object models. These tests are focused on evaluating the performance of 2D shape recognition systems in order to recognize 3D objects.

Chapter 3 is entitled “Active Recognition Models”. This chapter proposes a complete framework that will allow researchers to tackle the ambiguity problem for any generic object recognition system based on the active recognition paradigm. A mathematical background in which to implement different non stochastic methodologies, with the goal of reducing uncertainty, is presented here, and an in-depth comparative analysis of different models with which to implement the proposed framework is also included. The chapter ends by showing an experimental comparison with two well known existing frameworks.

Chapter 4 proposes a particular 3D Recognition System called CSS-NSSD. The principal contribution of the system lies in the object representation model. The objective is to normalize and reduce the object representation model with the aim of creating an active recognition system that is as efficient, accurate and fast as possible. We present the Canonical Sphere Section (CSS) as a solution that highly reduces a view-based object database by considering the objects’ non-redundant information. We also define the NSSD (Normalized Silhouette in the Fourier Spectral Dominium) descriptors to achieve a normalized shape representation of the object which is invariant to affine and geometrical transformations. The CSS-NSSD model has been experimentally tested and compared with other proposals. Firstly, the performance of the NSSD descriptor is compared with other popular shape descriptors, and secondly, the model is evaluated using four active recognition systems in an experimental robotic platform.

Chapter 5 presents a particular application of the proposed 3D object recognition system in a robotic setup. The robotic task consisted of grasping an isolated object in the scene, using the knowledge provided by the proposed 3D recognition system. The test was carried out on a 6 DOF robot with an on-board micro-camera and a two finger gripper. The chapter includes an explanation of the main stages in this process and shows the results obtained.
Chapter 6 is devoted to highlighting the main contributions of the thesis and presenting further improvement and future research lines.
View-Based 3D Object Recognition: An experimental comparative study

One of the most usual strategies for tackling the 3D object recognition problem consists of representing the objects by their appearance. 3D recognition can therefore be converted into a matter of 2D shape recognition. This chapter is focused on carrying out an in-depth qualitative and quantitative analysis of the performance of 2D shape recognition methods when they are used to solve 3D object recognition problems. Well-known shape descriptors (contour and regions) and 2D similarity measurements (deterministic and stochastic) are thus combined to evaluate a wide range of solutions.

2.1 Introduction

2.1.1 An Overview on Object Recognition

2D object recognition has been well researched, developed and successfully applied to many applications in industry. However, 3D object recognition is relatively new, and let us say that in general, 3D computer vision has only been developed over the last two decades. The main issue involved in 3D recognition is the large amount of information which needs to be dealt with. For example, 3D recognition systems have an infinite number of possible viewpoints, thus making it difficult to match the information provided by sensors in a knowledge database [16].

More particularly, much research has been carried out to develop algorithms in image segmentation and registration, or object recognition and tracking. Image segmentation, defined
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

as the separation of the image into regions, is usually the first step leading to image analysis and interpretation. Segmentation techniques can be classified into five groups [17]: threshold based, edge based, region based, classification (or clustering) based and deformable model based. As regards image registration techniques, they can be divided into two types of approaches: area-based and feature-based [18]. Area based methods compare two images by directly comparing the pixel intensities of different regions in the image, while feature-based methods first extract a set of features (points, lines, or regions) from the images and then compare the features. Area-based methods are often a good approach when there are no distinct features in the images. Feature-based methods are, however, often faster since less information is used for comparison. Feature-based methods are also more robust to viewpoint changes [19, 20], which often occur in vision-based robot control systems.

In general, object recognition approaches can be divided into two categories. The first approach utilizes the appearance features of objects such as their color and intensity. The second approach utilizes features extracted from the shape and matches these features in the database. One advantage of the feature-based approaches is their ability to recognize objects in the presence of lighting, translation, rotation and scale changes [18, 21]. The improvements that have been made to robots and vision systems signify that object recognition on board the robots are no longer limited to manufacturing environments. Some of the most representative vision-robot systems in which the recognition problem is tackled are shown as follows. Wong et al. [16] developed a system which uses spatial and topological features to automatically recognize 3D objects. A hypothesis-based approach is used for the recognition of 3D objects. In this system the best matching score is always used to determine the correct match, and is thus prone to false matches if the object in an image is not present in the database. Boker et al. [22] presented a system in which an industrial robot system is used for the autonomous disassembly of used cars, in particular, the wheels. The system uses a combination of contour, grey values and knowledge-based recognition techniques. The principal component analysis (PCA) is used to accurately locate the nuts of the wheels which were used for localization purposes. A coarse-to-fine approach is used to improve the performance of the system. The vision system was integrated with a force torque sensor, a task planning module and an unscrewing tool to form the complete disassembly system.

Some researchers have also simplified the recognition task through the use of methods which are invariant to scale, rotation, translation and partially invariant to affine transformation. These methods allow the object to be placed in an arbitrary pose. Jeong et al. [23] proposed a
method for robot localization and spatial context recognition. The Harris detector [24] and the pyramid Lucas-Kanade optical flow methods were used to localize the robot end-effector. The Harris detector and scale invariant feature transform (SIFT) descriptor [25] were employed to recognize spatial content. Peña-Cabrera et al. [26] presented a system to improve the performance of industrial robots working in unstructured environments. An artificial neural network (ANN) was used to train and recognize objects in the manufacturing cell. The object recognition process uses an image histogram and image moments which are fed into the ANN to determine what the object is. In Abdullah et al.’s work [2], a robot vision system was successfully used to sort meat patties. A modified Hough transform [27] was used to detect the centroid of the meat patties, which was then used to guide the robot to pick up individual meat patties. The image processing was embedded in a field programmable gate array (FPGA) for online processing.

Despite the fact that components such as camera calibration, image segmentation, image registration and robotic kinematics have been extensively researched, they exhibit shortcomings when used in highly dynamic environments. Real time self-calibration is a vital requirement of the system, while with image segmentation and registration it is crucial that the new algorithms will be able to operate under the presence of changes in lighting conditions and scale on blurred images. New robust image processing algorithms will thus need to be developed. Similarly, there are several approaches available to solve the inverse kinematics problem, these principally being the Newton-Raphson and Neural Network algorithms. However, these are hindered by accuracy and time inefficiency, respectively. It is thus vital to develop a solution that is able to provide high accuracy whilst simultaneously providing a high degree of time efficiency.

2.1.2 Main Problems in OR

The discussion shown until this point may suggest that computer vision is a fairly well understood subject which should find many applications. However, this is often far from true. There are many problems in computer vision that complicate viable solutions.

One of the most important technical problems is probably the choice of parameters in vision algorithms. These parameters are, for example, the scale for calculating derivatives and some thresholds used in image processing (for example edge detection). The optimal choice for such parameters greatly depends on the specific lighting conditions and on the scene. If the light, the color, the reflectance properties of the objects, or the number of objects in the
scene, etc. change, then a system with preset parameters will often yield bad results. Some techniques alleviate this problem by determining the parameters in a (semi-) automatic fashion [28, 29, 30], but these techniques rely on specific knowledge from examples, and the resulting choice of parameters is still only suitable for a limited set of situations.

Many object recognition systems store a large number of images that represent different object viewpoints in an efficient computer system memory (database). In order to improve the recognition process, the current view from the object to be recognized is compared with the database views, and the most similar view from the database is a solution. The goal, therefore, is to discover adaptable similarity measures for random variations of the environment such as:

- Viewpoint changes: Slight viewpoint variations induce changes in the view’s appearance.
- Photometric effects: Illumination variation and sensor distortion induce changes in the view’s appearance (color contour shape, etc.).
- Object position and occlusion: In some cases, the position of the objects in the scene changes, or an object is occluded.
- Background cluttering: Sometimes segmentation between the background and the query object is difficult because both have a similar color or texture.

Bearing all these possible random variations in mind, it is always possible to store all the object views for every possible circumstance, but at the expense of having a huge, inefficient database. The object representation model selected to reduce the database dimensionality is therefore very important.

Another problem to bear in mind is that images only provide 2D information about a 3D shape. A 2D image therefore frequently provides insufficient information with which to identify the object and correctly estimate its pose. Uncertainty and ambiguity problems frequently arise in such cases owing to the fact that no depth information is available. Different objects might therefore seem to be quite similar from different viewpoints, which affects the robustness of the 3D recognition system.

A more fundamental issue in computer vision concerns how prior knowledge is represented in a vision system. This prior knowledge consists of the model information from the objects and the imaging process. This knowledge should preferably be clearly visible in the system, or it should at least be clear what prior knowledge is assumed in each processing step. However,
in practice, much of this knowledge is implicitly encoded in the algorithms that are used. Let us, for example, consider an edge detection algorithm. The purpose of this algorithm is to detect changes in illumination, color or orientation from their appearance in the image. This means that the use of a certain algorithm implies assumptions about the way in which these changes will be visible in the image. The problem is that these assumptions are not usually obvious from the algorithm itself. Therefore, if the system is to be used in a situation in which these assumptions do not hold, it will not be a clear-cut job to adapt the system to this new situation.

Papers on computer vision research have also pointed out some problems with regard to aspects of theory and measurement [31, 32, 33]. It would appear that computer vision research lacks both a fundamental theory and a common experimental practice. There is no database of standard images from standard scenes and there are no quantitative measures for the quality of the results. Moreover, if a researcher wishes to test another algorithm on his images and compare the results directly, s/he usually has to implement the algorithm him/herself because the software is not available. One way in which to avoid this problems in practice consists of working on the theoretical side of the research. However, the assumptions that are often made in theoretical work make it hard to see how the results of this work will contribute to practical problems. In spite of the fact that people are aware of these problems, there does not yet appear to be a solution.

2.1.3 3D Object Recognition with Monocular Vision

Three-dimensional object recognition is the process of finding an object in a scene. This task implies determining the object’s identity and/or its pose (position and orientation) with regard to a particular reference frame. For instance, in object manipulation with robots, the pose of the object must be extracted through an accurate estimation of the translation and rotation parameters with regard to the robot coordinate system.

In the field of three-dimensional object recognition with monocular sensors, two main streams appear: view-based (or appearance-based) approaches and structural (or primitive-based) approaches. Primitive-based approaches yield a low performance when unexpected changes occur in the scene. However, view-based methods have become a popular representation scheme owing to their robustness to noise, photometric effects, blurred vision and changing illumination. The main advantage of this approach is that the image of the query object can be directly compared with a set of stored images in a database and it is efficient and robust to
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

variations in the scene. Indeed, the 3D problem has led to a 2D shape recognition question in which multiple views associated with the object from different points of views have to be handled. Each view in the database is thus associated with a particular viewpoint that corresponds with the current camera position (position and orientation). From here on, we shall use the term ‘shape’ to refer to the appearance of the object from a specific viewpoint - in a 2D context - and object’ as a general word to describe something in a 3D dimension environment. The 3D object pose estimation will signify geometric transformations between the camera position in the scene and the viewpoint from which the object is viewed in the database, whereas shape pose estimation will concern rotation, translation and scale in a 2D context.

Meanwhile, when a single view is taken to recognize an object, the principal problem is that one 2D image frequently provides insufficient information with which to identify the object and correctly estimate its pose. Uncertainty and ambiguity problems frequently arise in such cases owing to the fact that no depth information is available. In active recognition systems this handicap is addressed by moving the camera to different positions and processing several captures of the object until the uncertainty is resolved. Classical active recognition systems are made up of three main stages: the shape recognition algorithm - which concerns shape identification and shape pose estimation in a 2D context-, the fusion stage - in which the combination of the hypotheses obtained from each sensor position is carried out -, and the next-best-view planning stage - in which the optimal next sensor positions are computed-. The two last stages are used to improve the active recognition efficiency, thus reducing hypothesis uncertainty.

Since the view-based strategy converts 3D object recognition into a 2D shape recognition problem, an enormous amount of approaches concerning how to represent 2D shapes and how to measure similarities between shapes can be found in literature. However, to the best of our knowledge, no comparative study of different 2D shape recognition algorithms adapted to view-based 3D recognition systems has yet been reported. In order to provide a solution to this issue, the goal of this chapter is simply to carry out an in depth qualitative and quantitative analysis with regard to the performance of 2D shape recognition methods when they are used to solve 3D object recognition problems.
2.2 2D Shape Recognition for 3D Environments. State of the art

As was mentioned, the main stages in a 3D appearance-based object recognition system that uses one view are: shape representation, shape identification and pose estimation. Also, in this section we are going to analyze several methods from references to implement the main stages of a shape recognition system with special attention to the properties required by a 3D recognition system.

2.2.1 2D Shape Representation

2D shape representation is carried out through two principal descriptors: contour descriptors and region descriptors. Models based on contours are more popular than those based on regions. Contour-based methods need the extraction of boundary information which, in some cases, might not be available. However, region-based methods are more robust to noise and do not necessarily rely on shape boundary information, and they do not extract the features of a shape. The desirable characteristics of a shape representation model are:

1. Generality: The representation must be able to describe any shape and to preserve both global and local information.

2. Efficiency: The representation should be as simple as possible and should be obtained without excessive computational cost.

3. Robustness: Robust in cases of noise and occlusions in the image.

4. Stability: A stable set of features insures that small shape variations do not induce a noticeable change in the feature values [34] and this property gives robustness under small distortions of images.

5. Invariancy: The representation must be invariant to at least geometric transformations and, in the case of contour-based representations, invariant to the starting point.

Table 2.1 shows an overview of various 2D shape representation methods when applied to 3D recognition, and points out their advantages and drawbacks.

No matter what application is developed, all shape representation models share a common problem: object appearance can change drastically as the viewpoint changes owing to the transformation of the perspective. Most authors calculate changes in the viewpoint’s orientation
## 2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA descriptors</td>
<td>Region</td>
<td>Generality. Invariance: rotation, translation, scale, reflection.</td>
</tr>
</tbody>
</table>

Table 2.1: 2D shape representation methods and their characteristics.
2.2 2D Shape Recognition for 3D Environments. State of the art

through affine transformations. This may be an appropriate solution when the object is far away from the camera since slight shape distortions arise when the camera moves. Appearance-based object representations have different paths to achieve affine invariant properties: through the use of shape-normalization procedures, by using invariant shape similarity measures or by defining invariant shape descriptors.

A shape normalization method is an elegant pre-processing technique that transforms the distorted input shape into its corresponding normalized shape so that it is invariant under translation, scaling, skew, and rotation. Apart from the affine transformation parameters, to which the normalization is invariant, no other information is discarded. In fact, the normalization process consists of establishing an affine (linear) transformation that does not alter its original shape. A generalized normalization process with which to determine invariants is provided in Rothe et al. [48], and image normalization is tackled in Shen and Ip [37].

The second choice, concerning invariant similarity measurements, usually provides results of a low accuracy [49] [50] or high computational costs [51]. An approach with a similarity metric that is invariant to rotation, translation, and is scaling based has been proposed in Arkin et al. [52]. This work is applicable only to polygonal shapes.

Finally, many authors use invariant shape descriptors [53], [54]. Object recognition using normalized Fourier descriptors and neural networks has been presented in Wang and Cohen [55], while genetic algorithms for affine-invariant shape recognition have been proposed in Tsang [56]. Techniques based on the local features of curves include: grayscale local invariants based on the automatic detection of points of interest [57], affine invariants based on convex hulls for image registration [58], and local deformation invariants for contour recognition based on implicit polynomials [59]. Other approaches match two given contours by evaluating the affine parameters that maximize their similarity measure. This optimization is based on boundary moments [60] or Fourier descriptors [61]. They do not usually yield good results since, in most cases, part of the original curve is lost.

2.2.2 Similarity Measures

We shall refer to ‘identification’ as the process in which a query shape is classified in a database. This can be achieved through matching techniques or classification methods. Matching techniques have principally been developed for object recognition under several distortion conditions. Among the universe of classification techniques, we are interested in those that use similarity measures. This kind of methods is used in applications such as recognition in large
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean distance [62]</td>
<td>Deterministic</td>
<td>Efficiency</td>
</tr>
<tr>
<td>Hausdorff distance [63]</td>
<td>Deterministic</td>
<td>Invariance: rotation, translation, scale</td>
</tr>
<tr>
<td>Reflection distance [64]</td>
<td>Deterministic</td>
<td>Invariance: rotation, translation, scale, skew</td>
</tr>
<tr>
<td>City-block distance [64]</td>
<td>Deterministic</td>
<td>Efficiency</td>
</tr>
<tr>
<td>Cosine of the angle [65]</td>
<td>Deterministic</td>
<td>Efficiency</td>
</tr>
<tr>
<td>Bhattacharyya distance [66]</td>
<td>Stochastic</td>
<td>Robustness</td>
</tr>
<tr>
<td>Mahalanobis [67]</td>
<td>Stochastic</td>
<td>Invariance: scale</td>
</tr>
<tr>
<td>Bayessian [68]</td>
<td>Stochastic</td>
<td>Invariance: scale</td>
</tr>
<tr>
<td>k-nn [69]</td>
<td>Stochastic</td>
<td>Robustness</td>
</tr>
<tr>
<td>SVM [70]</td>
<td>Stochastic</td>
<td>Robustness</td>
</tr>
</tbody>
</table>

Table 2.2: Properties of several similarity measures.

image databases, where an image in the dataset “looks” very similar to the query image according to certain defined criterion. This criterion could be established by means of stochastic or deterministic methods. The selection of the correct hypothesis for deterministic methods uses a distance metric (Euclidean distance, norm $L_p$, Hausdorff’s distance, etc.), whereas the stochastic methods are based on probabilistic or statistics methods (bayesian networks, learning methods, etc.).

Most papers in literature argue that similarity measures should have the following properties:

1. Invariance: This is necessary when the shape descriptors are not invariant to geometric transformations.

2. Robustness and Sensitivity: Poor sensitivity leads to inadequate discrimination capability. Changes in the shape owing to noise must not affect the measure value.

3. Efficiency: The similarity values can be efficiently computed and compared.

Table 2.2 presents the most meaningful similarity measures, and shows both the type of criteria and the pros and cons for each one.

### 2.2.3 Shape Pose Estimation

Shape pose estimation determines the scale, translation and rotation parameters of the shape in a canonical reference system. Three pose methodologies can be used:
2.2 2D Shape Recognition for 3D Environments. State of the art

- Model based methods: The features of the representation model are used to estimate the pose. Examples of this strategy are geometric moments and Fourier descriptors.

- Learning based methods. The system is first trained by using a large set of images of the shape in different poses. A query shape is then identified and posed with regard to the training set (see [62]).

- Geometric algorithms. These are based on calculating the shape pose through geometric transformations between corresponding boundary points. One example in this field is ICP (Iterative Closest Point) based techniques.

We have chosen shape representation models (model based methods) to estimate the shape pose, since the performance of these methods tends to be high and does not require a large amount of training sets as in the case of learning methods. Models based on geometric transformations usually present low accuracy or the computational cost tends to be very high. Pose estimation based on shape representation models therefore offers the best trade between cost/accuracy parameters.

Many applications of 3D recognition systems such as object manipulation require a computed object pose with high accuracy in order to develop stable grasps. In this case, effectiveness during shape pose estimation is mandatory and the performance of the shape pose estimation model must be taken into account when designing the 3D recognition system.

2.2.4 Motivation of a Comparative Study

Figure 2.1 shows a general chart for a shape recognition system taken from the main modules and models discussed in the previous sections. In practice, most of the aforementioned shape representation methods can be combined with the similarity measure and pose estimation methods. However, the key question is: what is the optimal combination for a 3D recognition system?

Although most of the published shape recognition approaches show a good performance, it is not possible to decide which is the most appropriate since they have been tested in different databases, and probably under different conditions. Only a few references compare various methods under the same conditions and with the same databases [71, 72, 73, 74, 75, 76].

The principal issue here is that questions such as: what are the advantages/drawbacks of shape recognition systems using deterministic measure versus stochastic measures?, or what
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

![Figure 2.1: General scheme for a shape recognition system](image)

results are expected from a 3D recognition system that uses shape descriptors based on contour or in region?, have not yet been dealt with in literature, as is the case of the type of data (real or synthetic images) used in the recognition method and their influence on the final result. The first aim of this chapter is therefore to answer the aforementioned questions and to help future researchers to select robust and feasible shape recognition frameworks, specifically in the 3D recognition field.

The second point to consider is how to measure the performance of a 2D recognition system when it will be used for a 3D system. Traditional precision/recall analysis does not provide sufficient information to allow decisions about future strategies in a 3D recognition system to be made. For example, an ambiguity measurement in the 2D outputs would help the researcher to select evidence methods based on ambiguity to reduce uncertainty. In this chapter, we consider other evaluation methods focused on analyzing shape recognition output properties, such as the level of ambiguity among hypotheses.

2.3 Statement of the Experimental Tests

It is not easy to make an experimental comparison between different recognition methods since each one is tested under different conditions and with different databases. Moreover, the amount of details in each technique makes it impossible to reproduce the experiments in exactly the same way.
2.3 Statement of the Experimental Tests

2.3.1 Recognition Methods Set (RMS)

We have carried out a comparative study using a set of 2D shape representation models combined with a set of different similarity measures. From here on, we shall denominate all these methods as the Recognition Method Set (RMS). Of course, the selection of the chosen similarity measures and 2D representation models was not performed randomly. We chose only a few of the universe of methods that are available (some of which are referenced in Tables 2.1 and 2.2) after evaluating four aspects:

1. Significance: the method had to be highly referenced by others authors in important events and journals;

2. Reproducibility: it had to be possible for us to reproduce the method from the original paper;

3. Performance: There had to be a report proving a good performance of the method in the shape recognition field;

4. Completeness: the overall set of methods had to cover a wide spectrum of shape recognition strategies (as shown in Figure 2.1).

After considering the four criteria, we reached a consensus and selected a subset of representative techniques.

2D shape representation

- Fourier Descriptors (FD)
- Boundary Moments (HM)
- Integral Invariants (II)
- Shape Context (SC)
- Zernike Moments (ZM)
- Generic Fourier Descriptors (GFD)
- Complex Moments (CM)

Similarity measures
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

- Euclidian (ED)
- City Block (CB)
- Cosine of the Angle (C)
- Mahalanobis (MD)
- Battacharyya (BD)
- Support Vector Machine (SVM)

2.3.1.1 Computing shape descriptors

For all the images in a data set, we first extracted the image shape descriptors proposed in the RMS. For each shape descriptor, we experimentally found the lowest feature vector length that provided sufficient discriminatory power for our datasets. We used the following length of feature vectors for each shape descriptor:

**Contour-based**

In order to measure the similarities between two contour-based shapes using the methods from RMS, it is necessary for the feature vectors to have the same length. It is also possible to satisfy this constraint by applying a contour regularization process. For each shape descriptor, we experimentally found the lowest number of items (length) for each feature vector that provided sufficient discriminatory power for our datasets.

- Fourier Descriptors (64 descriptors): The contour has been regularized to 64 elements. After the regularization process, the Fourier Descriptors were computed following the same process as in [39].

- Boundary Moments (7 descriptors): The canonical 7 Boundary (Chen’s) moments were computed as in [37]. In this case it was not necessary to develop the contour regularization.

- Integral Invariants (64 descriptors): The contour was normalized to 64 points. The integral invariant was then computed as in [41] with radius $r = 0.15$

- Shape Context (7680 descriptors): The contour was regularized to 128 points. The shape context descriptor was then computed as in [42] with 12 angular bins and 5 radial bins.
2.3 Statement of the Experimental Tests

Region-based

- Zernike Moments (121 descriptors): We used the first 10 orders of Zernike moments, which were computed as in [44].

- Generic Fourier Descriptors (36 descriptors): For efficient shape description, only a small number of GFD features are selected for shape representation. In our implementation there are 36 GFD features reflecting 4 radial frequencies and 9 angular frequencies[45].

- Complex Moments (11 descriptors): The first 11 Complex Moments were taken [46].

In future, the number of features defined in this section will be referred as canonical number.

2.3.2 Experimental Platforms

We have implemented an object recognition system that is capable of running with two different experimental platforms. Platform 1 uses the well known Amsterdam Library of Object Images (ALOI) benchmark [77] and Platform 2 uses a 3DSL [78] database with 3D synthetic models in the dataset but with real images as sample images. Platform 1, based on the ALOI benchmark database, is an appropriate dataset with which to test the performance of a shape recognition system. The use of a benchmark dataset to validate the performance of any shape recognition represents an important advantage: the experimental results obtained can be validated by other researchers (using the same database). However, the ALOI dataset provides an incomplete object representation and the fact that the dataset does not include variant conditions in a real scene. This dataset could therefore be used to test shape recognition systems in controlled environments (e.g. industrial applications).

We have, therefore, also developed another test using a 3D synthetic library (3DSL) which allows us to take any view of an object (Platform 2). In 3DSL library, the geometrical models of the objects have been built in advance in our lab by using a VIVID 910 Minolta laser scanner sensor. Given a 3D model, we can define a set of homogeneous viewpoints over the nodes of a tessellate sphere and extract the depth image of the object from each view. We thus have a representation model which contains the complete information about the appearance of the object. This experimental platform is valid to test the performance of 3D recognition systems in non-controlled environments in which the test images are captured from a camera in a real scenario.
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

On both experimental platforms, the object contour is obtained after carrying out a thresholding process on the query image. In order to work under the same conditions in both databases and to achieve a better performance with the 3DSL database, the shapes have been normalized to skew deformations. In the case of contour-based approaches, (i.e. Fourier Descriptors (FD), Boundary Moments (HM) and Shape Context (SC)), Avrithiset.al’s method has been implemented [79]. For Integral Invariant (II) descriptors, the shift invariance was applied in order to achieve invariance to the starting point. However, for region-based descriptors (i.e. Zernike Moments (ZM), Generalized Fourier Descriptors (GFD) and Complex Moments (CM)), the skew invariance was obtained by using the Shen algorithm [80].

2.3.3 Evaluation Parameters

In order to evaluate the recognition results, we have defined three parameters, Hard Recognition Rate ($H_r$), Weak Recognition Rate ($W_r$) and Ambiguous Recognition Rate ($A_r$). The general idea here is that the proposed parameters can be used to study the uncertainty problem in a shape recognition process so that we can always decide the best strategy with which to solve this uncertainty. Whereas parameter $H_r$ simply provides a strict one-to-one recognition measure, $W_r$ and $A_r$ include softer alternative recognition measures. These parameters are defined in the following paragraphs.

1 Hard Recognition Rate ($H_r$) is:

$$H_r = \frac{n_e}{N} \cdot 100$$

2 Weak Recognition Rate ($W_r$) is:

$$W_r = \frac{n_e + n_p}{N} \cdot 100,$$

in which we have added the number of weak solutions $n_p$. We have a weak solution if the angle between the recognized and the true views is below a certain angular value $\epsilon > 0$. The Weak Recognition Rate parameter ($W_r$) should be used to implement strategies based on finding solutions which are close to the true solution.

3 Ambiguous Recognition Rate ($A_r$) is:
2.3 Statement of the Experimental Tests

\[ Ar = \frac{n_e + n_a}{N} \cdot 100 \]  \hspace{1cm} (2.3)

in which we consider the number of wrong solutions owing to ambiguity, \( n_a \). The explication of this term is as follows.

The idea is that although the shape is not specifically recognized within a view dataset, which implies a theoretical recognition failure in terms of parameter \( Hr \), it could be correctly classified into a group of similar shapes. If we generate the clustering of the original view dataset into a set of clusters, each containing similar shapes, and take into account the fact that the query view is classified in the correct cluster, a softer recognizing measure can be evaluated. The term \( n_a \) is therefore added in equation 2.3. In summary, \( n_a \) is the number of views wrongly recognized but that have been correctly classified in the set of clusters of similar views.

The power of parameter \( Ar \) then takes place inside the active recognition framework, specifically when the ambiguity problem arises in the recognition process. That is, if the shape is classified in the right cluster, the failure now somehow becomes a success. These cases, which frequently appear in 2D shape recognition, are considered in the definition of parameter \( Ar \). The clustering process was developed by using QT-clustering with a shape signature descriptor [10] as a characteristic vector.

Let us now provide a better motivation of the parameters presented above. The goal of the proposed parameters is to study the uncertainty problem and then decide the best strategy with which to solve the 3D recognition problem by following an active method. For example, if \( Hr \) yields low values, it is clear that other views of the object must be collected to reduce the uncertainty, but if \( Wr \) yields high scores, the use of viewpoints close to the current view could be sufficient to decrease the uncertainty. Parameter \( Ar \) scores the shape recognition capability to classify one view into a set of similar views so that high values signify that strategies based on the use of discriminative viewpoints could be effective. Active recognition systems based on the performance of the parameter \( Ar \) must concentrate the recognition effort on only a small number of hypotheses (clusters), the next best view being that which makes the uncertainty among all the hypotheses minimum. Furthermore, \( Ar \) is an indicator of the number of viewpoints (sensor positions) required to reduce uncertainty. Object recognition systems with a low \( Ar \) average imply the use of more viewpoints owing to the low the shape recognition system effectiveness.
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

2.4 Recognition Using Platform 1

2.4.1 Platform Setup

The ALOI-VIEW collection consists of 1,000 objects recorded under various imaging circumstances. More specifically, the viewing angle, illumination angle, and illumination color are systematically varied for each object. In our experiment, we have used a collection of objects which have been imaged from viewing angles spanning a range of up to 5 degrees. Figure 2.2 shows an example of an object represented from 72 viewpoints. RMS has been tested on 12 objects (see Figure 2.3). Note that the objects selected are very dissimilar free shapes.

In order to develop a learning process, we generate a set of images obtained from the prototype in the ALOI-VIEW collection for each object and viewpoint. By adding different noise levels and taking affine transformations on the original image we attempt to simulate external circumstances (illumination deficiency, lens blur,). in the scene. The main objective of this step is to accomplish the learning and simulate the efficiency of RMS in real scenes. The training set for one object and view is therefore composed of 15 images as follows: prototype image, two views close to the prototype, six images obtained after applying affine transformations to the prototype and six images with different Gaussian noise levels injected into the prototype image. Of these, six images are chosen for the training process and the others are used as test images. Figure 2.4 shows the training image set, while Figure 2.5 illustrates the effect that the noise produces on the contour after the image preprocessing.

After completing the learning phase, we ran RMS over 72 views with 3 images each. The three images per view are generated as follows: original image captured by the camera after applying an affine transformation and two images with added Gaussian noise (variance $\sigma_1 = 0.005$ and variance $\sigma_2 = 0.05$). Table 2.3 shows several examples of the images generated. A total of 2592 ($3 \times 72 \times 12$ objects) trials were thus carried out whose objective was to compare the different combinations between shape representation and shape identification algorithms in order to measure computational complexity and robustness in each situation.

2.4.2 Analysis of $Hr,Wr$ and $Ar$ Parameters

Table 2.4 allows us to provide an overview of the results and to make a comparison of the different combinations between shape representations and similarity measures. Each column in the table shows the recognition rates (in percentages) for each recognition parameter evaluated.
2.4 Recognition Using Platform 1

**Figure 2.2:** Object example from ALOI-VIEW collection viewed from 72 different viewpoints (figure from [45])

**Figure 2.3:** Objects used in the experimental tests.

Our conclusions are presented below. They are, on the one hand, split into contour and region based shape representations and are, on the other hand, split into deterministic and stochastic similarity measures, are presented below.
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

Figure 2.4: The training set for one object and view is: a) prototype image and two views close to the ground truth, b) six images obtained after applying affine transformations to the prototype and c) six images with different Gaussian noise levels injected into the prototype image.

Figure 2.5: Noise effects on the binary images after the image preprocessing
2.4 Recognition Using Platform 1

Table 2.3: Examples of test images.

<table>
<thead>
<tr>
<th>Without noise</th>
<th>$\varphi_1 = 0.005$</th>
<th>$\varphi_2 = 0.05$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Table 2.4: Comparative analysis of combinations of different shape descriptors and various similarity measures using ALOI dataset (Hard Recognition ($H_r$), Weak Recognition ($W_r$) and Ambiguous Recognition ($A_r$)).
2.4 Recognition Using Platform 1

Contour-based shape representation methods

- Fourier descriptors: Their robustness to noise is notable. In general, they perform best with deterministic measures and the Support Vector Machine. Most identification errors are achieved as a result of their invariance to reflection, and the Hard Recognition Rate ($Hr$) and Weak Recognition Rate ($Wr$) parameters have similar values. They are a good choice for 3D recognition systems dealing with ambiguous hypotheses owing to the good Ambiguous Recognition Rate ($Ar$) parameter results.

- Boundary Moments: Although this descriptor is very popular in 2D shape representation references, its performance in 3D recognition systems is lower than expected.

- Integral Invariants: Although in previous experiments the results when using test images without rotations (even with noise) are excellent, in 3D recognition systems (in which robustness to geometrical transformations is mandatory) the results using deterministic measures yield very poor scores. Only the $Wr$ values are acceptable with stochastic measures. These low rates occur as a result of the imprecision at the starting point. This is the method’s weakest key point.

- Shape Context: This is more robust than the Integral Invariants method, and the improvements that appear in the case of $Hr$, $Wr$ and $Ar$ parameters are very similar to those of Fourier Descriptors.

Region-based shape representation methods

- Zernike Moments: These are very sensitive to noise. Their performance is particularly low with City block distance.

- Generic Fourier Descriptors: Better results than Zernike Moments. They work quite well, especially when using Mahalanobis distance, but in general, their performance is not really notable.

- Complex Moments: These yield the best results even in added noise cases. Their best performance is achieved by using stochastic measures.

Deterministic similarity measures
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

- Euclidean distance: Its effectiveness depends on the descriptor’s robustness to noise. The combination with Fourier Descriptors and Complex Moments provides good rates, especially with \( Ar \) values.

- City block: This gives similar results to Euclidean distances. Nevertheless, the recognition rates fall when it is applied with region-based methods, except in the case of Complex Moments.

- Cosine of the angle: This provides similar results to those of Euclidean distance.

**Stochastic similarity measures**

- Mahalanobis distance: Poor recognition results (\( Hr \) and \( Wr \)) when using Mahalanobis distance, although meaningful improvements appear in the case of \( Ar \).

- Battacharyya: Similar results to the case of Mahalanobis distance. In general, it gives lower overall rates than the other methods.

- Support Vector Machine: This method achieves the highest rates and could be used in recognition systems that use the \( Wr \) and \( Ar \) evaluation parameters.

### 2.4.3 Computational Costs

One of the most important parameters used to evaluate the quality of a 3D recognition system is that of computational cost. Figures 2.6 and 2.7 show the execution time (in seconds) for representation models and similarity measures. The test has been run on a 1.3 Ghz Pentium IV computer. Figure 2.6 specifically shows the processing time needed to obtain the shape representation model, while the plots in Figure 2.7 correspond to the processing time needed to calculate similarity measures when considering two parameters: the number of descriptors used by the shape representation model (Figures 2.7(a) and 2.7(b)) and the number of views in the dataset (Figure 2.7(c)). In the last cases, the number of descriptors is 128.

Note that, as is clearly shown in Figure 2.7(a), deterministic measures, with the exception of the Cosine of the angle (C), show a better performance than stochastic measures. Figure 2.7(b) allows us to see the exponential behavior of the Mahalanobis distance function (MD) as the number of the descriptors in the shape representation model increases.

After analyzing the time rates according to the number of views in the dataset from 2.7(c), we can state that SVM and MD outperform with deterministic measures, the BD measure
is inadequate for large datasets and all the deterministic measures (Euclidean distance, City Block, Cosine of the angle) have similar computational costs.

### 2.4.4 Pose Estimation Accuracy

Another aspect which must be taken into consideration is that of pose estimation accuracy. We have used the quadratic mean error \((QME)\) between the shape sample \((S)\) and the shape identified in the dataset \((S')\) after being applied the transformation \((T)\) computed according to the representation model for Fourier Descriptors, Boundary Moments, Shape Context, Zernike Moments, Generalized Fourier Descriptors and Complex moments. In the case of the Invariant Integral method, the pose estimation is obtained through the contour normalization process parameters. Shapes are represented by \(Q\) couples of coordinates \((x_q, y_q)\) \(1 \leq q \leq Q\). Thus, let \((x'_q, y'_q)\) be the coordinates of \(S'\) and \((\hat{x}_q, \hat{y}_q)\) be the coordinates of the shape obtained with \(\hat{S} = T \cdot S\). Then

\[
QME = \frac{\sum_{q=1}^{Q} (\hat{x}_q - x'_q)^2 + (\hat{y}_q - y'_q)^2}{Q}
\]  

(2.4)

Table 2.5 shows the average of the quadratic mean error \((\text{Av}(QME))\) and the execution times required to compute the pose for different shape representation models. Among all the methods, the Fourier Descriptors yield the best accuracy and pose estimation calculation time. The contour normalization process with regard to the starting point is very inaccurate, and for this reason that the average of the quadratic mean error \((QME)\) is so high for Integral Invariant descriptors.
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

Figure 2.7: Processing time needed for similarity measures. Plots from a) and b) show the time rates according to the number of descriptors in the shape representation model. c) Times rates according to the number of views in the dataset for shape models using 128 descriptors.


2.4 Recognition Using Platform 1

<table>
<thead>
<tr>
<th></th>
<th>Av(QME)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD</td>
<td>0.1377</td>
<td>0.0230</td>
</tr>
<tr>
<td>HM</td>
<td>1.4671</td>
<td>0.0298</td>
</tr>
<tr>
<td>SC</td>
<td>1.9267</td>
<td>0.0330</td>
</tr>
<tr>
<td>II</td>
<td>13.221</td>
<td>0.0375</td>
</tr>
<tr>
<td>ZM</td>
<td>2.3306</td>
<td>0.0863</td>
</tr>
<tr>
<td>GFD</td>
<td>2.4992</td>
<td>0.0605</td>
</tr>
<tr>
<td>CM</td>
<td>2.4374</td>
<td>0.1079</td>
</tr>
</tbody>
</table>

Table 2.5: Average of the Quadratic Mean Error (Av(QME)) and execution time in pose estimation.

2.4.5 Object Identification Rates

So far, the experimental discussion in the previous sections has been focused on calculating the pose of the object in the scene. It is clear that estimating the pose of one object is a more restrictive task than recognizing the object within an object database. Obtaining the right pose through a 2D projection of the object may, in some cases, be a difficult and unsolved problem. For example, the pose estimation might be wrong as a result of symmetries or illumination effects. In order to compare the identification and pose results in our framework we have carried out a simple test. Figure 2.8 shows the object identification rate \( I_r \) achieved during the tests developed on Platform 1. In this case, we count the number of times that the RMS system successfully classifies the object.

Upon comparing results of \( I_r \) with the corresponding three parameters \( H_r, W_r \) and \( A_r \), which are shown in Table 2.4, we can appreciate notable differences between object identification and object pose results. We realized that the RMS system identifies the right object but that it sometimes chooses the wrong view. The successful score variation for \( H_r \), in other words \( |H_r - I_r| \), is between 7% and 34%, in which \( H_r < I_r \) in all cases. The score variation ranges for parameter \( W_r \) when compared with \( I_r \) is between 2% and 25%, in which \( W_r < I_r \) is also the case. As regards parameter \( A_r \), which considers similar views belonging to different objects, it is not surprising that the successful score variation has the contrary sign in this case. The range is thus between 3% and 32%, but now \( A_r > I_r \).
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

2.5 Recognition Tests Using Platform 2

2.5.1 Platform Setup

The objects belonging to the 3DSL dataset have been built in our lab. To do this, a high accuracy three-dimensional mesh model of each object was obtained in advance by means of a laser scanner sensor. Figure 2.9 presents a selection of objects from the 3DSL database. Note that the database is composed of both free and polyhedral shapes and even includes some similar objects. For instance, it would appear to be quite difficult to distinguish between objects 6 and 7.

As was previously mentioned, the set of tests that we have carried out on the 3DSL dataset are focused on evaluating the performance of RMS in non-controlled scenarios. In this case, the test image is taken from a camera in a real scene without special lighting requirements. In this section we shall focus on presenting the results for parameters $Hr$, $Wr$ and $Ar$. Since the processing time values are similar to those of the ALOI, no additional data and comments regarding computational cost are provided.

The 3DSL dataset is built by viewing the synthetic models from a set of homogeneous viewpoints and subsequently extracting the corresponding silhouettes from the projected images. These viewpoints are set by the lines from the vertexes of a tessellated sphere with 80 nodes to the object’s centroid. Figure 2.10 illustrates an object model inside the tessellated sphere, the projected image of the model and the depth-image from a specific viewpoint. In order to obtain the true object pose in the real scene (rotation with regard to the object in the

![Figure 2.8: Object identification rates (Ir).](Image)
2.5 Recognition Tests Using Platform 2

Figure 2.9: Samples of 3DSL Synthetic collection.

database), a set of images around the object are captured beforehand by the camera on-board the robot. Each image is then manually associated with its corresponding depth image in the database.

Figure 2.10: Image depth extraction. a) 3D object and the arrow corresponding to a selected viewpoint. b) Object model viewed from the selected viewpoint. c) Image depth.

The same training and testing procedure as that used in the ALOI dataset case is now followed. Thus, for each object, we generate 15 training images from each canonical view, which is defined by a node on the tessellate sphere. In this case, there are six training images corresponding to viewpoints around the node, which signifies "moving" the camera slightly but always maintaining the model’s centroid in the optic axis. The other training images are obtained by adding Gaussian, salt and pepper and sparkle noise to the canonical image. Figure
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

2.11 shows several training images of the dinosaur. The recognition test is accomplished by capturing one image from a camera located on the end-effector of a robot (Figure 2.12). We have taken a total of 86 images for this test. Some samples of this test data appear in Figure 2.13.

**Figure 2.11:** Training images from a node of the tessellated sphere. The first row shows images around a sphere node, the second row shows the image with Gaussian, salt and pepper and sparkle noise.

**Figure 2.12:** Experimental setup. The experimental platform uses a Stabli robot with a webcam on the end-effector. The robotic vision system captures images around the object. The robot positions correspond to nodes of an imaginary tessellate sphere centred in the scene.
2.5 Recognition Tests Using Platform 2

2.5.2 Analysis of $H_r$, $W_r$ and $A_r$ parameters

Table 2.6 shows a collection of plots including the recognition results for the original image (in the first column) and for the image with added Gaussian noise (second column). Hard Recognition Rate and Weak Recognition Rate values prove to have the worst performance in the majority of the methods. Only the Ambiguous Recognition Rate parameter presents acceptable rates, particularly for SVM. The reason for this is that we are using a test image in real conditions which implies introducing uncontrolled noise and variations in the input image. For instance, there are small variations in the true viewpoint with regard to the theoretical viewpoint in the database. On the other hand, the image is processed to obtain the contour and it is therefore slightly modified as a result of the image processing. All these factors make the query image appear to be deformed with regard to the best hypothesis in the database. The experimental test proves that although we have attempted to train the system by simulating possible shape deformations, the system behavior during the test is relatively hard to emulate using synthetic images.
Table 2.6: Comparative analysis of combinations of different shape descriptors and various similarity measures using 3DSL dataset (Hard Recognition ($H_r$), Weak Recognition ($W_r$) and Ambiguous Recognition ($A_r$))
Our conclusions, which are split into contour and region based shape representations and deterministic and stochastic similarity measures, are presented below.

**Contour-based shape representation methods**

- Fourier descriptors: Their recognition parameter rates generally show better results than the other contour descriptors in the case of the Ambiguity Recognition parameter, and maintain their robustness to noise. However, this is not recommended with Weak Recognition.

- Boundary Moments: Low performance in all cases.

- Integral Invariants: More sensitive to noise than Fourier Descriptors in the case of deterministic measures.

- Shape Context: Its good performance is notable, combined with stochastic similarity measures which maintain high rates in the case of the Ambiguous Recognition parameter.

**Region-based shape representation methods**

- Zernike Moments: Their results show high sensitivity to small shape variations.

- Generalized Fourier Descriptors: The improvement is not really notable.

- Complex Moments: they yield the best results even in the case of added noise. Their best performance is achieved using stochastic measures

**Deterministic similarity measures**

- Euclidean distance: Its performance decreases considerably (up to 10% for the Ambiguous Recognition parameter).

- City block: This gives similar results to those of Euclidean distance, showing a bad performance

- Cosine of the angle: We see similar results to those of Euclidean distance.

**Stochastic similarity measures**

- Mahalanobis distance: Poor recognition results ($H_r$ and $W_r$) when taking Mahalanobis distance. As regards $A_r$, it shows similar rates for all descriptors.
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

- Battacharyya: Similar results to those of Mahalanobis distance.
- Support Vector Machine: The best rates are achieved with this method when combined with Complex Moments, Shape Context or Fourier Descriptors.

2.5.3 Sensitivity to the number of views of the 3D representation model.

The last tests developed in this experimental platform are related to the shape recognition performance when the number of views used to represent the object is increased. As was mentioned previously, in the aforementioned analysis the objects are represented by 80 nodes. In this subsection we have therefore attempted to evaluate the effects on a shape recognition system when the objects are represented by views captured from a tessellate sphere with 320 nodes. We have repeated the tests developed in section 2.5.1, but using Dataset 3.

Table 2.7 compares the results for meshes with 80 and 320 nodes, showing the recognition rates for parameters $H_r$, $W_r$ and $A_r$. Note that, in summary, the recognition rates show slight variations in parameters $W_r$ and $A_r$, but that $H_r$ is approximately 15% lower with regard to the dataset with objects represented by 320 nodes owing to the uncertainty added by high similarities in the views closer to the correct solution.
Table 2.7: Comparative analysis of combinations of different shape descriptors and various similarity measures using 3DSL dataset (Hard Recognition (Hr), Weak Recognition (Wr) and Ambiguous Recognition (Ar)). The first row corresponds to Dataset with 80 views and the second row corresponds to 320 views (Dataset 3).
Figure 2.14 studies the relative variation of the recognition score for models of 80 and 320 nodes depending on the similarity measure and the kind of descriptors. Equation (2.5) represents the recognition mean relative variation for each parameter $P$ and similarity measure $S$, denoted as $\triangle \hat{R}(\hat{P},S)$, $\hat{R}_{srm,i}$ being the mean of the recognition score considering all the shape representation models for meshes with $i$ nodes.

$$\triangle \hat{R}(\hat{P},S) = \frac{\hat{R}_{srm,80}(P,S) - \hat{R}_{srm,320}(P,S)}{\hat{R}_{srm,80}(P,S)}$$ (2.5)

$P \in \{H_r, W_r, A_r\}$; $S \in \{EC, CB, C, MD, BD, SVM\}$

Equation (2.6) likewise formalizes the recognition mean relative variation for each parameter $P$ and shape representation model $M$, denoted as $\triangle \hat{Q}(P,M)$, $\hat{Q}_{srm,i}$ being the mean of the recognition score considering all the similarity measures for meshes with $i$ nodes.

$$\triangle \hat{Q}(P,M) = \frac{\hat{Q}_{srm,80}(P,M) - \hat{Q}_{srm,320}(P,M)}{\hat{Q}_{srm,80}(P,M)}$$ (2.6)

$P \in \{H_r, W_r, A_r\}$; $M \in \{FD, HM, II, SC, ZM, GFD, CM\}$

Part a) corresponds to $\triangle \hat{R}(\hat{P},S)$ for different parameters and similarity measures. Note that only parameter $H_r$ is really sensitive to the number of nodes on the model, with variation rates of up to 39% (MD case). Parameters $W_r$ and $A_r$ hardly vary the recognition percentages: 2% and 3% respectively. It is interesting to note that, contrary to the previous section, $A_r$’s variation is slightly higher than that of $W_r$’s. However, we can state that the alteration in the number of nodes in the representation model does not significantly affect recognition parameters $W_r$ and $A_r$. The same can be said for part b) of Figure 2.14 in which $\triangle \hat{Q}(P,M)$ is represented. In this case, most of the variation percentages for $W_r$ and $A_r$ are below 2% and 4% and parameter $H_r$ rises up to 62%. Descriptors HM (61.9%) and II (40.3%) again appear to be the most sensitive, whereas descriptor CM (19.8%) seems to be the most robust.

### 2.5.4 Sensitivity to the number of features

Another point to consider in this analysis is the relationship between the numbers of descriptors in the feature vector and the $H_r$, $W_r$ and $A_r$ results. Since the number of features for each particular representation method uses a different dimension, we have attempted to unify this analysis by showing the average recognition rates when we decrease or increase the number of features with regard to the canonical number defined in section . Thus, in Figure 2.15, for each shape representation and for parameter $H_r$, we provide the average rate of all the
2.5 Recognition Tests Using Platform 2

Figure 2.14: a) Recognition mean relative variation for meshes of 80 and 320 nodes for different similarity measures. b) Recognition mean relative variation for different 2D shape representation models.

similarity measures versus the relative number of descriptors compared with the canonical number imposed in section 2.5.4. In this case, we show only the plots for the $Hr$ parameter because the tendency is the same for the $Wr$ and $Ar$ parameters.

The results from Figure 2.15 prove that the canonical number of features, which corresponds to the value 100%, is optimal. Taking a minor number of features (which would correspond to values less below 100%) the average of $Hr$ clearly decreases, whereas, more items only maintain or decrease the $Hr$ results.

2.5.5 Comparison Results for Different Object Datasets

In addition to the analysis developed using the 18 objects from Figure 11, we have also experimented with two reduced datasets (5 objects) in order to discover the effect on the $Hr$, $Wr$ and $Ar$ parameters when a different database is used.

Dataset 1 includes dissimilar objects (objects 1, 3, 8, 14, 18), while Dataset 2 contains objects with some degree of similarity (objects 2, 9, 6, 7, 12). Experiments for these two datasets have been developed in the same manner as that explained in Section 2.5.1, although only the object related to the dataset in the test was used, and noise was not added to the input images.

Table 2.8 shows the experimental results for these two datasets. In this case, the intention of the analysis is to compare the shape recognition performance in a dataset with objects that share a similar appearance (Dataset 2) and a dataset with dissimilar objects (Dataset 1). Moreover,
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

**Figure 2.15:** Tendency of parameter $H_r$ versus the relative number of features with regards to the canonical number.

The aforementioned studies (Table 2.6) and those developed in this subsection have allowed us to discover the "sensitivity" of the shape recognition system to variations in the objects in the dataset.
Table 2.8: Comparative analysis of combinations of different shape descriptors and various similarity measures using two different 3DSL datasets. Dataset 1 includes dissimilar objects (objects 1, 3, 8, 14, 18). Dataset 2 has objects with some similarity (objects 2, 6, 7, 9, 12).
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

A comparison of the plots in Table 2.8 with the plots in Table 2.6 shows an increment in the $H_r$ and $W_r$ parameters values in Dataset 1. However, the $A_r$ parameter maintains similar rates. In the case of Dataset 2, the parameter $A_r$ also maintains similar rates but $H_r$ and $W_r$ parameters values decrease in most cases.

This variation in rates for the $H_r$ and $W_r$ parameters proves their dependence on the shape appearance in the dataset. If the objects are very different from each other, the shape recognition score/rate performance is very high and is affected only when shape descriptors are invariant to reflection. On the contrary, datasets with symmetrical objects and a similar appearance to each other achieve acceptable rates for the $A_r$ parameter only.

We shall now provide a more in-depth analysis, showing the relative variation of the recognition score for Datasets 1 and 2 depending on the similarity measure and the kind of descriptors. In order to make this document clearer, we shall maintain the nomenclature introduced in Section 2.5.3. Equation (2.7) represents the recognition mean relative variation for each parameter $P$ and similarity measure $S$, denoted as $\triangle \hat{R}(P,S)$, $\hat{R}_{srm,i}$ being the mean of the recognition score considering all the shape representation models for the $i-th$ database. Equation (2.8) likewise formalizes the recognition mean relative variation for each parameter $P$ and shape representation model $M$, denoted as $\triangle \hat{Q}(P,M)$, $\hat{Q}_{srm,i}$ being the mean of the recognition score considering all the similarity measures for the $i-th$ database. Figure 2.16 a) and b) presents the results for both cases $\triangle \hat{R}(P,S)$ and $\triangle \hat{Q}(P,M)$.

$$\triangle \hat{R}(P,S) = \frac{\hat{R}_{srm,1}(P,S) - \hat{R}_{srm,2}(P,S)}{\hat{R}_{srm,1}(P,S)} \quad (2.7)$$

$P \in \{H_r, W_r, A_r\} \land S \in \{EC, CB, C, MD, BD, SVM\}$

$$\triangle \hat{Q}(P,M) = \frac{\hat{Q}_{srm,1}(P,M) - \hat{Q}_{srm,2}(P,M)}{\hat{Q}_{srm,1}(P,M)} \quad (2.8)$$

$P \in \{H_r, W_r, A_r\} \land M \in \{FD, HM, II, SC, ZM, GFD, CM\}$

In general, we can state that parameter $H_r$ is the most sensitive to dataset changes, since it has a lower variation for parameters $W_r$ and $A_r$. Part a) clearly shows how MD and BD similarity measures achieve the maximum global variation, particularly in $H_r$ (62% and 58%) and $W_r$ (40% and 41%). For parameter $A_r$, lower and similar results are yielded for all similarity measures below 12%. In part b) we can see that the recognition score variation greatly depends on the descriptors for parameter $H_r$ (particularly high variations of 53% and 69% for HM and
2.6 Results Discussion and Conclusions

II). As regards parameter $W_r$, methods SC and CM (21% and 25%) are the least sensitive. Finally, all the shape representation methods provide a similar variation percentage, being around 10-12% for parameter.

Figure 2.16: a) Recognition mean relative variation for meshes of 80 and 320 nodes for different similarity measures. b) Recognition mean relative variation for different 2D shape representation models.

2.6 Results Discussion and Conclusions

This chapter presents a qualitative and quantitative study of the performance of a set of representative 2D shape recognition strategies when they are used as the pillar of 3D recognition solutions. In order to implement different recognition approaches, we have combined several of the most important 2D shape descriptors together with a set of deterministic and stochastic similarity measurements. Up to forty two combinations have been considered. The entire set of methods has been denominated as the RMS (Recognition Method Set).

The evaluation of the RMS is also focused on providing a quantitative criterion with which to develop strategies for uncertainty reduction in active recognition systems. The RMS is evaluated by means of three proposed parameters called Hard Recognition Rate ($H_r$), Weak Recognition Rate ($W_r$) and Ambiguous Recognition Rate ($A_r$). The evaluation of the RMS has also been extended to other parameters, such as computational cost and pose estimation accuracy.

Two different experimental platforms have been used during the tests to simulate the performance of RMS in controlled (Platform 1) and un-controlled (Platform 2) environments. The
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

Experimental results show that the performance of the shape recognition system varies according to the features of the experimental platform. After analyzing the results of this experimental study we can present the following summary of conclusions, which is divided into two sections.

- General remarks concerning shape representation models and similarity measures, no matter which experimental platform the 3D recognition system is performed on.
  - In general, appearance-based methods with a single view show acceptable recognition rates which decrease when external noise is added to the original images. Ambiguities caused by similarities between the appearance of different objects, symmetry factors and shape deformations owing to segmentation processes, illumination changes etc., increase the uncertainty of 3D recognition systems in real contexts.
  - Euclidean Distance and Cosine of Angle distances yield very similar results in all cases.
  - Shape Context, Fourier Moments and Complex Moments show good rates using SVM.
  - Although the Complex Moments show the best results, the execution time in this case is very high.
  - Since the 3D object recognition system might not require the accurate estimation of the object pose, Shape Context descriptors show good performance during the identification process for parameter \( W_r \), along with low execution times.
  - For large datasets, the Battacharyya similarity measure (BD) presents high computational costs. Meanwhile, the Manhalanobis measure (MD) is only recommended for shape representation models using few descriptors, owing to its exponential time rate behavior when the number of descriptors in the shape representation model is increased.
  - SVM techniques have the lowest computational costs for large datasets, but they depend on the number of descriptors in the shape representation model.
  - Fourier Descriptors are a good choice because, apart from their invariant properties, they require low computational time and are robust to noise.
2.6 Results Discussion and Conclusions

- Detailed remarks about the experimental review. Some observations and recommendations are now provided with regard to each of the platforms on which the 3D recognition will be implemented and the requirements of the 3D recognition application (computational costs, pose estimation accuracy).

- Platform 1 Strategies based on \( Wr \) and \( Ar \) parameters could be used under:
  *
  - High-accuracy pose estimation
    • Low computational cost: With regard to the \( Wr \) parameter, two choices are available: FD combined with CB or SVM. In the case of the \( Ar \) parameter, the same last choices are viable if ED is also added. The rates for \( Ar \) are better and more robust than those for.
    • Computational cost irrelevant: Complex Moments are definitely the best representation method combined with stochastic measures for \( Wr \) strategies or with any \( Ar \) strategies. In the last case, GFD shows a good performance when combined with SVM and Mahalanobis measures
  *
  - Low-accuracy pose estimation
    • Low computational cost: II and SC descriptors combined with SVM should be a good choice for \( Wr \) and \( Ar \) based systems.
    • Computational cost irrelevant: Apart from HM, all shape descriptors can be combined with SVM in \( Wr \) systems. GFD and deterministic measures are feasible for \( Ar \) systems.

- Platform 2 Low \( Wr \) rates for most of the RMS suggest that this strategy should be rejected, since only the CM descriptor shows acceptable values. The analysis with this database is thus focused solely on \( Ar \) strategies.
  *
  - High-accuracy pose estimation
    • Low computational cost: Fourier Descriptors with SVM are the best choice.
    • Computational cost irrelevant: CM and GFD descriptors with stochastic measures, ED and C deterministic measures improve other RMS.
  *
  - Low-accuracy pose estimation
    • Low computational cost: we recommend SC and SVM
2. VIEW-BASED 3D OBJECT RECOGNITION: AN EXPERIMENTAL COMPARATIVE STUDY

- Computational cost irrelevant: ZM should be combined with MD and BD distances.

None of the evaluation parameters \( (Hr, Wr \text{ and } Ar) \) recognize an object, but rather a particular view. In other words, they recognize a particular 2D shape. The recognition task is performed solely on a dataset composed of all available views of all the considered objects. Of course, in an active recognition system the object can be recognized when several particular views of the object are recognized. Therefore, rather than the concept object, which is not explicitly used in the document, the concept “shape” - meaning 2D shape - is used.

Although this chapter shows an exhaustive analysis of the performance of 3D systems using one view, it is important to bear in mind that the selection of the strategies with which to develop any 3D recognition system must consider the kind of objects in the dataset, which signifies a degree of similarity between them. Moreover, of the results obtained for our dataset, only \( Ar \) strategies yield reliable results. But these strategies only associate a view to a cluster of silhouettes. For a realistic 3D object recognition problem, several views must therefore be taken to reliably identify an object, and the best next view algorithm may play a fundamental role in the entire recognition process.
3

Multiple View-Based 3D Object Recognition. A framework for active recognition systems

As it was reported in Chapter 2 reports, the information contained in a single image of the scene might be insufficient to identify and pose an object in the scene. This chapter defines a complete framework to tackle the ambiguity problem for any generic object recognition system based on the active recognition paradigm. Although most methods deal with uncertainty using stochastic models, we have developed a mathematical background to implement different non stochastic methodologies with the goal of reducing the uncertainty.

3.1 Introduction

The use of monocular vision to develop 3D object recognition introduces serious uncertainty problems. The uncertainty signifies that one specific view of an object might be similar to other views of the same, or even different, objects. In the case of comparing images using features or descriptors extracted from the object, some factors such as noise, illumination variations and segmentation errors may corrupt the outputs of the feature detector. A well known strategy that solves the ambiguity problem is based on using multiple views of the object. Thus, 3D active recognition paradigm proposes a planning algorithm that sequentially locates the sensor at the most convenient position and orientation with the aim of minimizing the uncertainty.

The essential stages in the active recognition process can be found in Figure 3.1. We can
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

distinguish three principal modules: the 3D object representation, the shape recognition and the active strategy. As the 3D object representation models and the shape recognition modules have been discussed in the previous chapter, a brief review about active strategy is developed in the following section.

![Figure 3.1](image)

**Figure 3.1:** General scheme of an active recognition system based on monocular vision.

**Active strategy.**

The objective of an active strategy is focused on solving the uncertainty problem by using the set of hypotheses obtained in the previous stages. This is usually driven by two subprocesses: information fusion and sensor planning. The output of the shape recognition stage is a set of hypotheses which represents potential views of candidate objects belonging to the database. The goal of the information fusion process is to define an evidence function which measures the goodness of the new hypotheses in a sequential and accumulative process in which the camera must be moved to the theoretical next best position. The objective of the sensor planning process is therefore to choose the next sensor position according to a selection criterion.

Based on the 3D main modules of an active recognition system described above, Figure 3.1 shows a generic scheme of a 3D active recognition system in which the image of the scene is processed and the hypotheses are established by means of the shape recognition module. The new hypotheses are then fused to the results obtained from previous steps and a termination criterion, which concerns the uncertainty, is finally evaluated. If the termination criterion is
3.2 Previous works in Active Strategies in 3D Object Recognition Problems

not verified, a new sensor position is calculated. The system will repeat all further active steps until the object uncertainty is bellow a given threshold.

In this chapter we are particularly interested in how to define and manage the uncertainty problem in an active framework. The following sections are therefore devoted to presenting a brief review of the most representative active strategies that focus on 3D recognition.

3.2 Previous works in Active Strategies in 3D Object Recognition Problems

An active recognition strategy can be thought of as a process which associates the history of the observations made of the scene with the next best view to be taken. Given a history of observations, the selection of the new view must be made through a criterion which measures how useful the chosen view is under uncertainty terms. The methods which deal with uncertainty can be classified as follows:

- Methods based on traditional probability theory.
- Methods based on neo-calculi techniques. For example, amongst those techniques used to represent uncertainties there are fuzzy logic based methods, confidence factors and Dempster-Shafer calculus.
- Heuristic methods. Here the uncertainties are not modeled under explicit notations but are embedded in domain-specific procedures and data structures.

In active recognition systems, a great number of researchers use probabilistic models to tackle the uncertainty problem [81, 82, 83, 84, 85]. All this research is based on the good statistical properties of the PCA descriptors. However, in some cases, PCA based approaches are not recommended to represent the object views. For example, a 3D object recognition approach which is applied to an object manipulation task (i.e. grasping) demands high accuracy during the object pose estimation stage. In this case, the PCA descriptor is not able to satisfy this requirement since the shape pose estimation does not guarantee an acceptable accuracy. As regards active strategies based on stochastic models, they have the drawback of requiring large datasets during the training process in order to learn the best action (the best sensor position in our framework). Moreover, if a new object is added to the dataset, the entire system must be retrained.
As far as we know, just a few number of active recognition systems have been implemented using shape descriptors without stochastic properties. Moreover, most of the proposed active strategies are probabilistic and depending of the feature properties used and the dataset. A set of the most representative active strategies based on non-stochastic models is presented in the following paragraphs.

Gremban and Ikeuchi [86] present a scheme for planning multiple views in an object recognition task. They use Aspect-Resolution Trees which are built on the basis of aspect diagrams with the aim of planning multiple observations and thus recognizing the object in the scene. The authors show results for a vision-based sensor using a haptic sensor. They give examples of object recognition based on sensors with the aim of detecting specularities and properties of the objects. These specularities determine the aspects of the object. They consider that this matter is challenging since a single image of a specular object yields very little information, specularity being a local phenomenon. Moreover, many different object poses can yield the same pattern of specularities. The authors show recognition results with only three objects. This system has the particularity of being independent of the shape features.

Liu and Tsai [87] describe a multiple view-based 3-D object recognition system. They use two cameras and extract both silhouettes of the object, which is located on a turntable. The system first reduces the ambiguity by taking images from a top view. This allows the shape of the top view to be normalized, the object centroid to be established and the principal axis of the object to be aligned. The system takes a side view of the object, analyzes its silhouette features and then rotates it by 45 degrees. This procedure is repeated until the object is recognized. The system has been implemented using geometric features.

Hutchinson and Kak [88] use an aspect graph to represent information about the objects in their model base. They present a system that dynamically plans the sensor position based on the current best estimated position. They propose an automatic movement of the sensor, and then determine the maximum ambiguity which would remain if the operation were applied. The system then selects the operation which minimizes the remaining ambiguity. They use the Dempster-Shafer theory to combine evidence and analyze the proposed operations. Only simple objects are used in their experiments.

Kovacic et al. [89] propose a general framework to solve the ambiguity problem based on grouping similar views in a feature space (a cluster). The system learns the changes in the grouping from each possible action and records the action which maximally separates the views originally belonging to the same cluster. Doing this for all clusters allows a complete
recognition-pose-identification plan to be pre-compiled. A tree structure encodes the next best view relative to the current one during the object recognition. The main drawback of this framework is that it is not very robust to errors during the classification process. Thus, if the cluster selection is wrong in any step, the system fails.

On similar lines to Kovacic’s work, Gonzalez et al. [90] developed an active recognition system using the information related to the clustering process. The difference between both works lies in the sensor planning process. In Kovacic et al. [89], the sensor movement for each dataset cluster is computed during an off-line process. This system presents low robustness because if the system selects a wrong cluster, the recognition process fails. However, in the system proposed in [90], this problem is solved by planning a new sensor position on-line. This strategy is more robust but implies a higher computational cost. The idea of using a view-sphere as a structure which is able to study the dissimilarity between different views of objects on-line and to select to next sensor position appears for the first time in this work.

Considering the handicaps of the active recognition systems in the state of art, we have design a framework to solve most of those problems. Moreover, our framework satisfy the follow constraints:

1. Flexibility to design the active recognition system since it is independent of the shape recognition model and the active strategy employed.

2. Recognition of ambiguous objects.

3. Accuracy during object pose estimation without to increase the number of nodes in the sphere from which captures the object’s appearance. The object pose is computed using the geometric properties of the shape descriptor model.

4. Computational efficiency using the minimum number of sensor positions to reduce uncertainty.

5. An on-line sensor planning model allowing it to be more robust to errors during the estimation of hypothesis.

6. Explicit inclusion, in the framework model, the robot workspace. In this way, other objects could appear in the scene.
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

3.3 Active Recognition Framework

3.3.1 Shape representation

Consider a set of \( N \) objects \( O = \{ o_1, \ldots, o_N \} \). Let a generic object \( o_i \) have a synthetic model which is viewed from different view directions corresponding to the nodes of a tessellated sphere of \( J \) nodes: this sphere has a radius of \( r \), the object is placed in its center, and the image is captured in the direction defined by the node with the camera pointing to the center of the sphere. These nodes are labeled with integer positive numbers \( j \) following a given order structure. Let us associate the view \( j \) of the object \( i \) with a synthetic image \( I_{i,j} \), \( 1 \leq i \leq N, 1 \leq j \leq J \).

Consider the sphere shown in Figure 3.2. The position of node \( j \), which is representative of the view \( j \), is given in spherical coordinates by a pair of angles \( (\psi_j, \sigma_j) \), \( \psi_j \) and \( \sigma_j \) being the azimuth and elevation coordinates respectively. The position of node \( j \), expressed in the absolute cartesian coordinate frame attached to the sphere (see Figure 3.2), is then given by

\[
\vec{u}_j = (\cos(\sigma_j) \cos(\psi_j), \cos(\sigma_j) \sin(\psi_j), \sin(\sigma_j))^T
\] (3.1)
a unity vector. A frame is also associated with each node, given by three mutually orthogonal unit vectors shown in Figure 3.2: \( \vec{x}_j, \vec{y}_j \) and \( \vec{z}_j = -\vec{u}_j \). Note that all vectors will be expressed as columns from now on.

Let us associate the view \( j \) of the object \( i \) with a feature vector \( \vec{v}_{i,j} \), \( 1 \leq i \leq N, 1 \leq j \leq J \). Feature vector \( \vec{v}_{i,j} \) refers to boundary coordinates or region coordinates corresponding to the object as it appears in the image \( I_{i,j} \). Moreover, the components of these feature vectors usually depend on the object and its pose. Let us define an operator \( \Phi(\vec{v}_{i,j}) = \vec{w}_{i,j} \) that transforms \( \vec{v}_{i,j} \) into another feature vector \( \vec{w}_{i,j} \), where the dimension of \( \vec{w}_{i,j} \) is usually lower than the dimension of \( \vec{v}_{i,j} \), which is invariant with regard to the pose (this includes invariance to image rotation, translation and scaling). The elements of feature vector \( \vec{w}_{i,j} \) can be principal components, Fourier descriptors, shape histograms, moments, etc. Let us carry out this transformation for all the feature vectors \( \vec{v}_{i,j} \), \( 1 \leq i \leq N, 1 \leq j \leq J \) in the database.

Let us then define a view database \( B = \{ b_l \} \) (\( 1 \leq l \leq L \), where \( L = N \times J \)), and where each element \( b_l \) includes:

1. The object label \( i_l \).
2. The view label \( j_l \).

3. The synthetic image of object \( i_l \) taken from node \( j_l \) : \( I_l \).

4. The feature vector \( \tilde{v}_l \) of the synthetic image \( I_l \).

5. The invariant feature vector \( \tilde{w}_l \) of image \( I_l \).

6. A scalar value \( X_l \) representing the level of ambiguity.

7. A scalar that represents the accumulated evidence function \( E_l \).

8. A column vector of six components which contains the accumulated Euler representation vector \( T_l \).

The meaning of the last two items will be explained later.

The scalar value \( X_l \) representing the “level” of ambiguity is a useful parameter which is focused on active recognition systems using information about the ambiguity that is present among dataset views. This “level” of ambiguity could be interpreted as a measure of how different or discriminative a view is from the rest of the dataset views. This level of ambiguity could be computed by means of clustering methods [91]. The clustering algorithms split the dataset in a set of groups in which each group contains a set of coarse similar views. The \( X_l \) value could therefore be computed as:
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

\[
X_l = \frac{1}{\text{ord}(\Omega^l)}
\]  

(3.2)

where a view \( b_l \) belonging to a highly voted cluster is then more ambiguous than another which belongs to a poorly voted group. This idea is synthesized in equation (3.2). The function \( \text{ord}(\Omega, l) \) computes the number of views in cluster \( \Omega \) where the view \( l \) have been associated. This equation provides the ambiguity score of the \( l \)-th component of the dataset, \( \Omega^l \) being the cluster to which this belongs.

Note that in our active recognition system, label \( i_l \) of \( b_l \) will allow the object to be recognized, and label \( j_l \) will allow the object’s pose (position and orientation) to be calculated, according to an algorithm that will be developed in Subsection 3.3.4. In association with this database, we also define the function \( l = \mu(i_l, j_l) \), which allows the view and image feature vector of object \( i_l \) seen from view \( j_l \) to be retrieved from the database.

3.3.2 Shape recognition

Let us assume that we have taken the view number \( k \) of the object. The image obtained is denoted as \( I^{(k)} \), which has a feature vector \( \vec{v}^{(k)} \). The goal of the shape and pose recognition problem is to find the vector \( \vec{v}_l \) in the database which is most similar to \( \vec{v}^{(k)} \).

Let us define a generic dissimilarity measure \( D(\vec{a}_1, \vec{a}_2) \) between two feature vectors \( \vec{a}_1 \) and \( \vec{a}_2 \), both of which are of the same arbitrary dimension. Vectors \( \vec{a}_1 \) and \( \vec{a}_2 \) could be feature vectors \( \vec{v} \) or invariant feature vectors \( \vec{w} \) in dependency of the dissimilarity measure selected. Since \( \vec{v} \) is variant to geometric transformations, the dissimilarity measure \( D \) must be invariant, such as Hausdorff distance or Reflection distance. If \( D \) is based on invariant feature vectors \( \vec{w} \) as an argument, then it is possible to use similarity measures such as: Manhattan distance, Cosine of the angle, City block, the \( L_2 \)-norm, etc. \[92\]. Since \( \vec{w} \) is given by \( \vec{w} = \Phi(\vec{v}) \), the dissimilarity function will from now on be refereed by using \( \vec{v} \) as arguments.

Let us apply the above measure to determine the dissimilarity in the shapes between the object viewed in \( I^{(k)} \) and the synthetic image \( I_l \). We then calculate \( D(\vec{v}^{(k)}, \vec{v}_l) \) between the scene invariant features vector and the invariant feature vector of the synthetic image of the database. Moreover, a dissimilarity threshold \( \epsilon \) is defined in association with these measures, which allows us to determine the set \( B^{(k)} \) of database views that contains shapes that are similar to the shape of the measured image \( I^{(k)} \) by using the expression:
3.3 Active Recognition Framework

\[ B(k) = \{b_l\}, \ b_l \ | \ D(\vec{\nu}(k), \vec{v}_l) \leq \varepsilon \ \land \ b_l \in B \] (3.3)

The pose of image \( I^{(k)} \) with regard to the image \( I_l \) associated with the element \( b_l \) of \( B^{(k)} \) is denoted as \( p^{(k)}_l = (\phi^{(k)}_l, \vec{d}^{(k)}_l, \lambda^{(k)}_l) \), the elements of that vector being the rotation angle, the translation (it is a vector with two coordinates), and the scaling. Let the algorithm that determines this pose be denoted as \( P(I^{(k)}, I_l) \). Another measure function is defined in order to measure the dissimilarity between a set of \( n \) vectors \( D(\{\vec{a}_1, \vec{a}_2, ..., \vec{a}_n\}) \).

In order to score the ambiguity between a set of \( I_1, I_2, ..., I_n \) of \( n \) views, a function \( A \) is defined that takes into account the dissimilarity among the feature vectors of the views \( D(\{v_{l_1}, ..., v_{l_n}\}) \), in addition to the level of ambiguity of each view \( \{X_{l_1}, ..., X_{l_n}\} \). Function \( A \) can be based on Bayesian theory, fuzzy logic, euclidean distance, etc.

We shall now define the evidence function \( E(\vec{\nu}(k), b_l) \). Let us assume that we evaluate the hypothesis that the first view taken by the sensor corresponds to \( b_l \) by means of moving the sensor and taking further views. The evidence function therefore measures the possibilities that the image \( I^{(k)} \) taken from a further view (\( k^{th} \) view) - and whose feature vector is \( \vec{\nu}(k) \) - might correspond to the object and pose represented by \( b_l \), and its particular form depends on the recognition strategy used: probabilistic, possibilistic, heuristic or deterministic.

3.3.3 Sensor displacement model

A 6 degrees of freedom robot is used in order to carry out the sensor movements in our active recognition system. The relationship between the position and coordinate frame at the tip of the robot (where the sensor is located) and the position and coordinate frame at the base of the robot is defined by the kinematic model of the robot. This is obtained as a product of homogeneous transforms which depend on the robot geometry and the joint variables. We denote as \( \Theta = (\theta_1, ..., \theta_6)' \) the vector that includes the six joint variables. Since a robot has six revolute joints, all the joint variables are angles. Let us denote the total homogeneous transform as \( R^b_tT(\Theta) \), which allows us to express a vector given in the robot tip frame (Rt) in terms of the robot base frame (Rb).

Consider that the sensor (or equivalently the tip of the robot) is pointing at an object \( i_l \) with a direction represented by the node \( j_l \). Moreover, assume that there is similarity between the corresponding synthetic image \( I_l \) of object \( i_l \) at node \( j_l \) and the image \( I^{(k)} \) taken by the sensor, so that the pose between these two images can be determined: \( p^{(k)}_l = (\phi^{(k)}_l, \vec{d}^{(k)}_l, \lambda^{(k)}_l) = P(I^{(k)}, I_l) \).
Angle $\phi_l^{(k)}$ describes the rotation of the sensor/tip robot frame in relation to the $j_l$ node frame about $\vec{z}_{j_l}$. Vector $\delta_l^{(k)}$ describes the translation of the sensor/tip robot frame in relation to the $j_l$ node frame in the plane defined by axes $\vec{x}_{j_l}$ and $\vec{y}_{j_l}$. This is expressed as $\vec{d}_l^{(k)} = (x_l^{(k)}, y_l^{(k)})'$, whose components represent the displacement in each of these two axes. Finally, assume that the scaling factor $\lambda_l^{(k)}$ allows us to determine the distance $d_l^{(k)}$ from the sensor origin (once it has been centered by having corrected the displacement $\vec{d}_l^{(k)}$) to the center of the tessellated sphere.

The homogeneous transform from node $j_l$ frame ($S_n$) to the frame in the center of the tessellated sphere ($S_b$) is given by:

$$S_b^{S_n}(j_l) = \begin{bmatrix} \vec{x}_{j_l} & \vec{y}_{j_l} & -\vec{u}_{j_l} & r\vec{u}_{j_l} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

(3.4)

the homogeneous transform from the robot tip frame ($R_t$) to the node $j_l$ frame ($S_n$) is given by:

$$S_n^{R_t}(\phi_l^{(k)}, \vec{d}_l^{(k)}, d_l^{(k)}) = \begin{bmatrix} \cos(\phi_l^{(k)}) & -\sin(\phi_l^{(k)}) & 0 & 0 \\ \sin(\phi_l^{(k)}) & \cos(\phi_l^{(k)}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & \delta_l^{(k)} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r - d_l^{(k)} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

(3.5)

and the homogeneous transform from the robot base ($R_b$) to the frame in the center of the tessellated sphere ($S_b$) is

$$S_b^{R_b} = S_n^{R_t}(\phi_l^{(k)}, \vec{d}_l^{(k)}, d_l^{(k)}) \cdot S_n^{S_a}(j_l) \cdot S_n^{R_t}(\phi_l, \vec{d}_l, d_l^{(k)}) \cdot S_b^{R_b}(\Theta).$$

(3.6)

where $S_b^{R_b}(\Theta) = S_b^{R_b}(\Theta)^{-1}$. Note that matrix $S_b^{R_b}$ is invariant because it relates two fixed frames: tessellated sphere origin and robot base. This can be calculated from (3.6) by using the measures $\Theta$ and $p_l^{(k)}$, and by assuming that $l^{(k)}$ corresponds to the view $b_l$. Its inverse matrix $S_b^{R_b}$ expresses the object position and orientation with regard to the robot base frame if hypothesis $b_l$ is verified, i.e. the pose of the object with regard to the robot base frame. This matrix will be used in what follows, and will hereafter be denoted as $S_b^{R_b} = T_l$, where the index $l$ indicates the hypothesis under which it is obtained.
3.3 Active Recognition Framework

Let us assume that we now wish to move the sensor to the direction represented by node $m$, but we also wish to maintain the distance $d_l^{(k)}$ from the center of the tessellated sphere. The frame corresponding to this node is therefore expressed by the homogeneous transform

$$S_{b}^{T}(m) = \begin{bmatrix} \bar{x}_m & \bar{y}_m & -\bar{u}_m & r\bar{u}_m \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

(3.7)

and the homogeneous transform corresponding to the sensor frame is, after some calculations:

$$S_{R}^{T}(m, d_l^{(k)}) = \begin{bmatrix} \bar{x}_m & \bar{y}_m & -\bar{u}_m & d_l^{(k)} \bar{u}_m \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$ 

(3.8)

We thus obtain

$$T_l^{-1} = S_{b}^{T}(m, d_l^{(k)}) \cdot R_{R}^{T} T(\Theta_m)$$

(3.9)

where $\Theta_m$ is the robot joint variables vector needed to reach the direction of node $m$. Equation (3.9) yields the robot homogeneous transform

$$R_{R}^{T} T(\Theta_m) = T_l \cdot S_{b}^{T}(m, d_l^{(k)})$$

(3.10)

from which vector $\Theta_m$ can be obtained by carrying out the robot inverse kinematic transform.

Let us then define the function $\Theta_{m,l} = \Psi(m, d_l^{(k)}, l, T_l)$ which provides the robot joint variables needed to move the sensor to a position that points to the center of the tessellated sphere from a distance $d_l^{(k)}$ with a direction defined by node $m$, under the hypothesis that $b_l$ is verified (it is the database object that matches the image that has been viewed last), and $T_l$ is the pose of the object corresponding to that hypothesis (which has been calculated). This function can be calculated by substituting the result of (3.8) in (3.10), operating this last equation, and then carrying out the inverse kinematic transform.

Let us also define the inverse of the previous function $m = \Psi^{-1}(\Theta, d_l^{(k)}, l, T_l)$ in the sense that it provides, for a given hypothesis $l$, an object pose $T_l$ obtained under the hypothesis of having viewed $b_l$ with a certain sensor position, and a new specified sensor position (given by $\Theta$), the node $m$ of the tessellated sphere which is associated with the view of the object that is most likely to be seen by the sensor. This function is calculated as follows:

1. Calculate the homogeneous transform that expresses the sensor frame and position ($R_t$) in terms of the tessellated sphere frame ($S_b$):
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

\[ \frac{S_b}{R_t} T = T_i^{-1} \frac{R_b}{R_t} T(\Theta) \]  

(3.11)

2. Obtain the roll unity vector of the sensor frame \( \vec{z}_s \): 

\[ \vec{z}_s = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \cdot \frac{S_b}{R_t} T \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \]  

(3.12)

3. Label \( m \) represents the node whose associated position unity vector is most aligned with \( -\vec{z}_s \). It is thus the node that yields the maximum value of the scalar product: \( -u_j \circ \vec{z}_s \), \( 1 \leq j \leq J \).

Finally, recall that a frame can be expressed in reference to a base frame by means of its Euler angles and the position of its origin (all of which refer to that base frame). We denote these Euler angles as \( \alpha \) (roll), \( \beta \) (pitch), and \( \gamma \) (yaw), and the relative position of the frame origin as \( \vec{\kappa} \). These are expressed in a compact vectorial form as

\[ \mathcal{T} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \vec{\kappa} \end{bmatrix} \]  

(3.13)

The transform that passes from the previous Euler representation to the homogeneous transform is denoted as \( T = \Upsilon(\mathcal{T}) \), and the inverse transform is denoted as \( \mathcal{T} = \Upsilon^{-1}(T) \). The mathematical expressions of both direct and inverse transforms can be found in standard robotics textbooks (e.g. [93]).

3.3.4 Active recognition

The method used to recognize an object by means of active sensing is defined as follows. This process must be carried out in several steps (several sensor movements). A criterion with which to choose the next sensor location (position and direction) that yields the image that allows the minimization of the ambiguity among the view candidates (hypotheses) must then be developed. The sensor direction is chosen from among those defined by the nodes of the tessellated sphere.
3.3 Active Recognition Framework

The active recognition strategy framework proposed here proceeds in an iterative manner. At each step, new view candidates are obtained, and a decision model is built that allows us to determine what the next best view should be. This model has the structure of a tessellated sphere, is denoted as the D-Sphere model, and is obtained as described below.

First an initialization is carried out which includes:

1. Accumulated evidence functions $E_l$ are initialized to zero for all database elements $b_l, 1 \leq l \leq L$.

2. Accumulated Euler representation vectors $T_l$ are initialized to zero for all database elements $b_l, 1 \leq l \leq L$. These will allow us to obtain the homogeneous transform that expresses the object position and orientation with regard to the robot base frame (object pose) under verification of hypothesis $b_l$.

3. The sensor is moved to an arbitrary position and direction, but pointing to the center of the object. The distance to the object’s center is unknown. The vector of the robot joint variables corresponding to that location is $\Theta^{(-1)}$.

4. Capture the image $I^{(-1)}$, taken from the direction associated with $\Theta^{(-1)}$.

5. Process this image and obtain the features vector $\vec{v}^{(-1)}$.

6. Obtain the invariant features vector $\vec{w}^{(-1)}$ of image $I^{(-1)}$.

7. Determine the best view candidate $b_*$ as the database element that minimizes the dissimilarity measure $D(\vec{v}^{(-1)}, \vec{w}_l), 1 \leq l \leq L$.

8. Calculate the pose between the image $I^{(-1)}$ and the image corresponding to the hypothesis $b_*: p_*^{(-1)}$.

9. Estimate the homogeneous transform matrix $T_*^{(-1)}$ that expresses the object position and orientation with regard to the robot base frame under verification of hypothesis $b_*$ and the sensor position $\Theta^{(-1)}$. This is carried out by calculating (3.6), with $\Theta^{(-1)}$, the previously obtained pose, and the node $j_*$ corresponding to the hypothesis view $*, (l = *)$.

10. A distance of $d$ between the sensor and the center of the object (center of the tessellated sphere) is defined.
11. The next sensor position $\Theta^{(0)}$ is calculated as the position that is pointing to the center of the object from the same direction as in the previous position $\Theta^{(-1)}$, but is now at a distance of $d$ from the object’s center. From now on, the subsequent robot movements will be carried out by maintaining this distance between the object and the sensor.

The proposed iterative active recognition process is explained as follows. Let us assume that we have completed the iteration step $k-1$. As result of this step, we have obtained: 1) the hypothesis database $B$ with updated accumulated evidence (the evidence accumulated in hypothesis $l$ after $k-1$ steps is denoted as $E_{l}^{(k-1)}$), and with updated accumulated Euler representation vector $T_{l}^{(k-1)}$ (this will allow us to estimate the poses of all the objects and nodes with regard to the robot base); 2) the robot variables $\Theta^{(k-1)}$ corresponding to the next sensor location. The algorithm then starts from $k = 1$, and performs the step $k$ from step $k-1$ by carrying out the following three tasks in a sequential manner.

3.3.4.1 Task 1: Obtaining new view candidates

The actions involved in this task are the following:

1. Move the sensor to position $\Theta^{(k-1)}$.
2. Capture the image $I^{(k)}$, taken from the direction associated with $\Theta^{(k-1)}$.
3. Process this image and obtain the feature vector $\vec{v}^{(k)}$.
4. Obtain the invariant feature vector $\vec{w}^{(k)}$ of image $I^{(k)}$.
5. Determine the set $B^{(k)}$ of view candidates (or hypotheses) by using (3.3) and $\vec{v}^{(k)}$.
6. Estimate the homogeneous transform matrices $T_{l}^{(k)}$, $\forall l \mid b_{l} \in B^{(k)}$. These express the object position and orientation with regard to the robot base frame under verification of hypothesis $b_{l} \in B^{(k)}$ and sensor position $\Theta^{(k-1)}$. This is carried out by calculating (3.6), with $\Theta^{(k-1)}$, the pose between the image $I^{(k)}$ and the hypothesis image $b_{l} \in B^{(k)}$: $p_{l}^{(k)}$, and the node $j_{l}$ corresponding to the hypothesis view $l$.
7. Calculate the evidence measures of all the views of $B^{(k)}$:

$$E_{l}^{(k)} = E(\vec{v}^{(k)}, b_{l}), \quad b_{l} \in B^{(k)}$$

(3.14)
8. Determine, for each view candidate $b_l \in B^{(k)}$, the view that would be seen of the object corresponding with the sensor placed at $\Theta^{(0)}$ if this view $b_l$ (seen with the sensor placed at $\Theta^{(k-1)}$) were true. This is achieved by using function $\Psi^{-1}$, which allows us to determine the most likely node of the tessellated sphere that is seen when the sensor moves to a position $\Theta$ around an object. This calculation is carried out in two stages: 1) determine the most likely node to be seen from sensor position $\Theta^{(0)}$:

$$m_l = \Psi^{-1}(\Theta^{(0)}, d^{(k)}_l, I, T^{(k)}_l), \quad \forall b_l \in B^{(k)}, \quad (3.15)$$

2) determine the corresponding view of object $i_l$ from that node:

$$\hat{i}_l = \mu(i_l, m_l), \quad b_l \in B^{(k)} \quad (3.16)$$

9. Update the accumulated evidence values of all the views of $B$. Only the evidence of hypotheses $\hat{i}_l$, which are those corresponding to the candidate views $b_l \in B^{(k)}$ obtained with the sensor at the position $\Theta^{(k-1)}$, are reinforced by increasing their accumulated evidence. The other view hypotheses remain unchanged. It is then necessary to carry out.

$$E^{(k)}_l = \mathcal{K}(E^{(k-1)}_l, E^{(k)}_l), \quad \forall b_l \quad (3.17)$$

The evidence of each object at a given step $k$ is thus the accumulated evidence of all the steps performed with images taken from the current and previous sensor positions.

10. Update the accumulated Euler representation vectors associated with the candidate views:

$$T^{(k)}_l = T^{(k-1)}_l + E^{(k)}_l \cdot \Upsilon^{-1}(T^{(k)}_l), \quad \forall b_l \quad (3.18)$$

The vector which is added to $T^{(k)}_l$ is weighted by its evidence value, obtained when looking at the object from position $\Theta^{(k-1)}$.

Figure 3.3 illustrates the new view candidate obtaining process
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

Figure 3.3: New view candidates obtaining process. A set $B^{(k)}$ of view candidates (or hypotheses) is determined for an image $I^{(k)}$ captured at step $k$ and the accumulated evidence $E_{i}^{(k)}$ and accumulated Euler representation vectors $T_{i}^{(k)}$ are updated for each of the candidate views $b_{l} \in B^{(k)}$.

3.3.4.2 Task 2: Determination of the decision model

The decision model is built by carrying out the following actions:

1. Determine the best view candidate $b_{*}$, between the database elements of $B^{(k)}$, that yields the minimum value in the dissimilarity measures: $D(\vec{v}^{(k)}, \vec{v}_{l})$, $b_{l} \in B^{(k)}$.

2. Estimate the sensor positions (given in terms of robot joint variables) that correspondingly point at the tessellated sphere center for each node $m$ at a distance $d$ if hypothesis $b_{*}$ were verified:

$$\Theta_{m,*}^{(k)} = \Psi(m,d,*_{*},T_{*}^{(k)}), \quad 1 \leq m \leq J \quad (3.19)$$

Note that $T_{*}^{(k)}$ has already been calculated in Step 6 of Task 1.

3. Determine the nodes that correspond to the set of sensor positions $\{\Theta_{1,*}^{(k)}, \ldots, \Theta_{J,*}^{(k)}\}$ obtained previously, under each of the hypotheses considered in this step (each element $b_{l}$ of database $B^{(k)}$):
3.3 Active Recognition Framework

\[ j_{m,l} = \Psi^{-1}(\Theta_{m,*}^{(k)}, d, l, T_l^{(k)}), \quad \forall l \mid b_l \in B^{(k)} \]  
(3.20)

where \(1 \leq m \leq J\). Note that again matrices \(T_l^{(k)}\) have already been calculated in Step 6 of Task 1.

4. For a given sensor position \(\Theta_{m,*}^{(k)}\), estimate the view that would be expected if hypothesis \(b_l\) were verified. This estimation is obtained from the database \(B\) by taking into account that the object label is \(i_l\), and the node label is \(j_{m,l}\), which was calculated in the previous action. This view is therefore \(h_{m,l} = \mu(i_l, j_{m,l})\), and the expected features vector is \(\vec{v}_{hm,l}\). After repeating this estimation process for all the hypotheses of \(B^{(k)}\), we form the set of feature vectors of the images taken by the sensor if it were placed at \(\Theta_{m,*}^{(k)}\) under the different hypotheses: \(\{\vec{v}_{hm,l}\}, \forall l \mid b_l \in B^{(k)}\).

5. Obtain the level of ambiguity of the previous vector set: \(\rho_m = A(\{h_{m,l}\})\) where \(h_{m,l}, \forall l \mid b_l \in B^{(k)}\) and repeat this for all the sensor positions \(1 \leq m \leq J\).

6. Finally, determine the reachability of each of the \(\Theta_{m,*}^{(k)}\) calculated, i.e., if the corresponding robot tip/sensor location is inside the robot workspace and there is no obstacle in this location. We define a binary function \(\varsigma = \varsigma(\Theta)\), which gives a value of 1 if the location corresponding to \(\Theta\) is reachable and 0 if not. This function defines the reachability in the robot configuration space. We thus have \(\varsigma_m = \varsigma(\Theta_{m,*}^{(k)}), 1 \leq m \leq J\) for the nodes of the tessellated sphere. This function is calculated only once at the beginning of the active recognition process because a static environment has been assumed. In the case of having a dynamic environment in which obstacles alter their positions over time, this function should be recalculated at each step of the iterative process in accordance with the new location of the obstacles.

Note that, as a result of the previous process, we have obtained an auxiliary tessellated sphere which is aligned with the sphere corresponding to hypothesis \(b_\ast\). We associate with each node \(m\) of this new sphere: 1) the level of ambiguity among the feature vectors obtained under all the hypotheses of \(B^{(k)}\) if the sensor is placed to point towards the object from that node, \(\rho_m\); 2) the corresponding sensor location expressed in robot joint variables, \(\Theta_{m,*}^{(k)}\); 3) the reachability condition of the node, \(\varsigma_m\).
This sphere is denoted as the *decision sphere*, and will hereafter be referred to as the *D-Sphere*, because it contains the most significant information needed to make the decision of where to move the sensor in the next step.

### 3.3.4.3 Task 3: Next best view decision

Finally, the next best view sensor position is determined from the *D-Sphere*. The following actions are involved.

1. Calculate the “motion effort” of transporting the sensor from location $\Theta^{(k-1)}$ to a location associated with the node $m$. This “motion effort” is given by a function

   $$ g_m = g(\Theta, m) \quad (3.21) $$

   which is calculated by assuming that $\Theta = \Theta^{(k-1)}$, for $1 \leq m \leq J$. This function may express robot energy consumptions, the time required to move the robot from one location to the other, the distance from the initial to the target sensor frames, or simply the norm of the difference between initial and target joint variables.

2. Obtain the node $m_o$ that maximizes the *decision function*:

   $$ \beta(\rho_m, g_m, \varsigma_m), \quad 1 \leq m \leq J \quad (3.22) $$

   over the *D-Sphere*. This function may be of the form

   $$ \beta(\rho_m, g_m, \varsigma_m) = \varsigma_m \cdot \beta'(\rho_m, g_m) \quad (3.23) $$

   where $\beta'$ is a non decreasing function in $\rho_m$ and a decreasing function in $g_m$. Note that structure (3.23) makes the $\beta$ decision function 0 in the case of unreachable sensor locations.

3. The robot joint variables corresponding to the best next sensor location are then obtained from the *D-Sphere* as $\Theta^{(k)} = \Theta_{m_o}$.

   The iteration process ends when a view hypothesis $\hat{l}_o$ reaches a given level $\eta$ of accumulated evidence: $E_{\hat{l}_o}^{(k)} \geq \eta$. Element $b_{l_o}$ of the database $B$ is then considered to be the right
3.3 Active Recognition Framework

hypothesis and it gives the recognized object $i_o$. Moreover, the pose of this object with regard to the robot base is obtained from

$$T_{b_o} = \hat{\mathbf{J}}^{(k)}_{b_o} \left( \mathbf{X}^{(k)}_{b_o} \right)$$

which gives an estimation of the pose that is calculated from an average of the weighted Euler representation vectors obtained from each sensor position.

Figure 3.4 shows the process used to build the D-Sphere and the next best view selection.

![Diagram of the D-Sphere and view selection process](image)

**Figure 3.4**: Process used to build the D-Sphere and the next best view selection. a) Show for the best view candidate $b_*$, the sensor positions that correspond to pointing towards the centre of the tessellated sphere through each node $m$ at a distance $d$ if $b_*$ is verified. b) D-Sphere estimation. The level of ambiguity among the views mapped in the node is computed for each D-Sphere node. c) Next best view selection based on a decision function $\mathcal{J}(\rho_m, g_m, \varsigma_m)$

Figure 3.5 illustrates with an example the proposed framework performance in case of uncertainty by ambiguity. The first view of the scene ($k = 1$), yields five candidate views
belonging to two different objects. For each candidate is computed their evidence value ($E^{(k)}$) and the accumulated evidence ($\mathcal{E}^{(k)}$). The D-Sphere model is built and the next sensor position is computed. Then, the sensor is moved ($k = 2$) and a new scene image is captured. Again, the evidence value is estimated and the accumulated evidence updated. Finally, the accumulated evidence of one candidate ($o_{1,16}$) satisfies the threshold ($\eta$) and the recognition process stops.

Figure 3.5: Example of the proposed framework performance in case uncertainty by ambiguity.

As a consequence of all the previous considerations, we have described our proposal for an active recognition system framework, which exhibits a fixed overall structure, but which permits the implementation of different algorithms and criteria in some specific functions which are embedded in that structure:

1. Function $\vec{w}^{(k)} = \Phi(\vec{v}^{(k)})$ is used to obtain the pose invariant feature vector.
3.4 Active Strategies Methodologies

2. The dissimilarity measure $D(\vec{a}_1, \vec{a}_2)$ between two vectors.

3. The dissimilarity measure among a set of $n$ vectors $D(\{\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n\})$.

4. The evidence function $E(\vec{v}^{(k)}, b_l)$ of an hypothesis $b_l$.

5. The level of ambiguity between a set of views: $A(\{h_{m,l}\})$.

6. The motion effort function $\xi(\Theta, m)$ of (3.21).

7. The decision function $\beta(\rho, g, \xi)$ of (3.22), or $\beta'(\rho, g)$ of (3.23).

This framework also needs to settle two thresholds: $\varepsilon$ in (3.3), and $\eta$ to finish the process.

3.4 Active Strategies Methodologies

The following section provides a detailed explanation of various methodologies which can be used to develop active strategy models based on the D-Sphere framework. Let us assume that we are at step $k$ of the active strategy. We will therefore compute the evidence function $E(\vec{v}, b_l)$ and the level of ambiguity $\rho_m$ according to the possibilistic ($\Pi$), heuristic ($H$) or deterministic ($\Delta$) methodologies.

3.4.1 Possibilistic method ($\Pi$)

Possibility theory is a mathematical theory which deals with certain types of uncertainty as an alternative to probability theories. The possibility theory was introduced by Zadeh in 1978 as an extension of his theory of fuzzy sets and fuzzy logic [94].

Possibilistic frameworks can be implemented by means of: experts’ experience, probabilistic and heuristics information. In this work, we have chosen heuristic information in order to allow object shapes to be represented by descriptors without stochastic properties. The heuristic information is therefore provided by the $X_l$ element from a dataset view $b_l$.

The possibility of each hypothesis will be denoted as $\Pi(b_l|v^{(k)})$ and is computed by taking into consideration the dissimilarity measure between the hypothesis and the scene view ($D(\vec{v}_l, \vec{v}^{(k)})$):

$$\Pi(b_l|v^{(k)}) = X_l \land D(\vec{v}_l, \vec{v}^{(k)}), \forall l \mid b_l \in B^{(k)}$$  (3.25)
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

where $\&$ is a t-norm operator. We have used the minimum function as a t-norm operator, the membership function for $X_l$, $D(\bar{v}_l, \bar{v}^{(k)})$, and the output are Gaussian functions.

The evidence function could then be computed as:

$$\Pi E^{(k)} = \Pi(b_l\mid X_l)$$

and according to the possibilistic model, the accumulated evidence for each hypothesis must be updated by:

$$E^{(k)} = K(E^{(k-1)} \cdot E^{(k)}) = E^{(k-1)} \& \Pi E^{(k)}$$

Finally, the $\Pi \rho_m$ score for each D-Sphere node $m$ is computed by using the following function:

$$\Pi \rho_m = \Pi A(h_{m,l}) = X_{h_{m,l}} \& \ldots \& X_{h_{m,q}}$$

where $q$ is the number of hypothesis in step $k$.

Note that in this case, the ambiguity function $A$ does not depend on the dissimilarity among the view feature vectors, and depends solely on the ambiguity level of each of the views mapped onto a D-Sphere node.

### 3.4.2 Heuristic method ($H$)

Heuristics models are typically used when there is no known method with which to find an optimal solution under either the given constraints (of time, space etc.) or at all [95]. The proposed heuristic model is in complete analogy with the possibilistic method. The heuristic model is therefore based on the information collected by $X_l$. Functions $H E^{(k)}_l$, $E^{(k)}_l$ and $H \rho_m$ are computed as follows:

$$H E^{(k)}_l = [(1 - D(\bar{v}_l, \bar{v}^{(k)})) \cdot X_l \mid b_l \in B^{(k)}]$$

$$E^{(k)}_l = K(E^{(k-1)}_l \cdot E^{(k)}_l) = E^{(k-1)}_l + H E^{(k)}_l$$

$$H \rho_m = H A(h_{m,l}) = min X_{h_{m,l}}$$
3.5 Experimentation

Note that $H_p_m$ is computed again using only the levels of ambiguity of the views mapped onto the D-Sphere node, and gives as output the worse case: the level of the most ambiguous views.

3.4.3 Deterministic method ($\Delta$)

A deterministic algorithm is an algorithm which, in informal terms, behaves predictably. Given a particular input, it will always produce the same output. The evidence of hypotheses is thus computed by means of the dissimilarity function as:

$$\Delta E^{(k)}_l = 1 - D(\vec{v}_l, \vec{v}^{(k)})$$

and the accumulated evidence is updated as follows:

$$E^{(k)}_l = K(E^{(k-1)}_l, E^{(k)}_l) = E^{(k-1)}_l + \Delta E^{(k)}_l$$ (3.33)

According to the deterministic concept, we compute $H_p_m$ as a function which measures the dissimilarity between the feature vectors of the views mapped onto each node $m$ in the D-Sphere. The score is defined by the worst case which corresponds to the minimum distance between a pair of feature vectors. It is given by:

$$H_p_m = A(\{h_{m,l}\}) = D(\vec{v}_{h_{m,l}})$$ (3.34)

where $D$ gives the distance between the closest pair of vectors of $h_{m,l}$.

Observe that in this case function $A$ is based on the dissimilarities between object candidates, while possibilistic and heuristic methodologies also consider the level of ambiguity with other dataset views not belonging to $h_{m,l}$.

3.5 Experimentation

This section provides a qualitative and quantitative analysis of the proposed framework by means of a set of experimental tests. The experimental tests have been divided into two experiments. The first (Experiment1) is focused on developing an in-depth analysis of the recognition rate and recognition costs parameters when the framework is implemented by different feature vectors, using the active strategies from Section 3.4. The second experiment (Experiment2) compares the performance of an active strategy based on the proposed framework with other
active recognition systems from references considering datasets with: 1) a large number of objects; 2) ambiguous objects and 3) non ambiguous objects. This comparative analysis is developed using images captured from a real environment (robotic platform) and syntectic images from the syntectic dataset objects.

The Experiment1 is focused to prove the flexibility of the proposed framework to the shape recognition model and active strategy, meanwhile, in Experiment2 using different datasets in the robotic platform is evaluated the computational efficiency; the capability to deal with ambiguous objects and the effectiveness of the planning model. The experiments with the syntectic images provide a quantitative analysis about the accuracy during object pose estimation.

The performance of the proposed active framework will be evaluated through the use of two parameters: recognition rate ($R$) and recognition costs ($C$). This last parameter is computed by the follow equation:

$$C = W \cdot (T_m + T_s)$$

where $W$ is the average number of sensor positions, signifying the number of steps used by the active recognition system, $T_m$ is the average time taken to move from one position to the next and $T_s$ is the computational cost of one iteration step (the processing time of the shape recognition plus the time rate in the active strategy).

### 3.5.1 General statement of the experimental tests

The experimental setup consisted of a Staübli RX90 robot with a camera on the end-effector of the robot. This vision-robot system was able to capture images around the object placed in the scene. Figure 3.6(a) illustrates a typical scene with an isolated object placed on a table inside the robot workspace.

The tests were carried out on a 3D Synthetic Model library (3DSL) [78]. Each object model of the 3DSL library was built in advance by means of a VIVID 910 Minolta laser scanner sensor. The final result was a set of high accuracy three-dimensional meshes which synthesize the objects that were placed further away in the experimental setup.

The experiments were carried out by selecting 18 free-form objects from this library (3DSL). This set of objects can be observed in Figure 3.6(b) and are denoted as Dataset 1. Two different sets of objects from Dataset 1 were selected to build another two invariant datasets. Dataset 2 includes dissimilar objects (objects 1, 3, 8, 11, 17), while Dataset 3 contains objects with
3.5 Experimentation

some degree of similarity (objects 6, 7, 9, 10, 15). The object selection was carried out in order to study the proposed framework’s performance with large datasets, datasets with ambiguous objects and datasets with dissimilar objects. The knowledge database is built by first taking the projected images of the synthetic models from 80 viewpoints which are set by the position of the nodes in an imaginary sphere that wraps the model (Figure 3.6(c)). Figure 3.6(d) shows several samples of images captured in the vision-robot system. Note that the scene illumination alters according to the robot arm position.

![Figure 3.6: Experimental setup. a) The experimental platform uses a Staubli robot with a camera on the end-effector. b) Synthetic objects in Dataset 1. c) Object model placed in the tessellated sphere and some views. d) Samples of images captured in the vision-robot setup and used during the active recognition tests.](image)

3.5.2 Experiments 1

This experiment is focused on studying the proposed active recognition framework for different shape recognition methods and the proposed active strategies using Dataset 1. We selected the Euclidean distance [62] and feature vectors: Fourier Descriptors (FD)[39], Boundary Moments (HM)[43], Principal Component Analysis (PCA) [96], Zernike Moments (ZM)[44] and Complex Moments (CM) [46]. All of these feature vectors are invariant to geometric transformations. FD, HM and PCA are contour-based, while ZM and CM are region-based. It is also possible to evaluate the framework performance by using different shape representation models with different computational costs and performances (number of hypotheses).

The selection of the shape representation models that were chosen was neither easy nor random. We were only able to choose a few of the universe of methods that are available in the bibliographic references after evaluating four aspects: 1) Significance: the method was highly referenced by others authors in important events and journals; 2) Reproducibility: the
method could be reproduced by us from the original paper; 3) Performance: There was a report proving a good performance of the method in the shape recognition field; (4) Completeness: the overall set of methods covered a wide spectrum of shape recognition strategies (as shown in 3.1). After taking these four criteria into consideration, a consensus was reached and a subset of representative techniques was selected.

Table 3.1 shows a collection of methods used to implement the proposed framework.

<table>
<thead>
<tr>
<th>( \vec{v} )</th>
<th>shape silhouette, shape region</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \vec{w} = \Phi(\vec{v}) )</td>
<td>( \vec{w} = FD [90], \vec{w} = BM [43], \vec{w} = PCA [96], \vec{w} = ZM [44], \vec{w} = CM [46] )</td>
</tr>
<tr>
<td>( D(\vec{v}_l, \vec{v}(k)) )</td>
<td>( \sqrt{\Sigma(\vec{w}_l, \vec{w}(k))^2} )</td>
</tr>
<tr>
<td>( \mathcal{D}({v_l}) )</td>
<td>( \min(D(\vec{v}<em>{h</em>{m,t}}, \vec{v}<em>{h</em>{m,p}}), 1 \leq t, p \leq q, t \neq p, q ) is the number of hypothesis</td>
</tr>
<tr>
<td>( E_{l}^{(k)} )</td>
<td>( \Pi E_{l}^{(k)}, H E_{l}^{(k)} \Delta E_{l}^{(k)} )</td>
</tr>
<tr>
<td>( \rho_m )</td>
<td>( \Pi A \cdot H A \cdot \Delta A )</td>
</tr>
<tr>
<td>( B^{k} )</td>
<td>( \arg{D(\vec{v}_l, \vec{w}(k)) } &lt; \varepsilon )</td>
</tr>
<tr>
<td>( P(\vec{w}_l, \vec{w}(k)) )</td>
<td>( \vec{w} = FD [90], \vec{w} = HM [97], \vec{w} = PCA [98], \vec{w} = ZM [99], \vec{w} = CM [46] )</td>
</tr>
<tr>
<td>( g_m = \mathcal{G}(\Theta^{(k)}, \Theta_m) )</td>
<td>( \sqrt{\Sigma(\Theta^{(k)}, \Theta_m))^2} ), where ( \Theta ) are the cartesian coordinates of ( \Theta ) robot join variables</td>
</tr>
<tr>
<td>( \zeta_m )</td>
<td>Configured off-line considering whether the node ( m ) is accessible by the robot end-effector</td>
</tr>
<tr>
<td>( J )</td>
<td>( \max_m(\rho_m \cdot g_m \cdot \zeta_m) )</td>
</tr>
</tbody>
</table>

Table 3.1: Collection of methods used to implement the proposed framework

Figure 3.7 shows an example of the membership functions used in the possibilistic method.

![Figure 3.7](a) Membership function for \( X_{X_l} \). (b) Membership function for \( D(\vec{w}_l, \vec{w}(k)) \). (c) Output membership function

Affine invariance was achieved by applying a shape normalization process. The goal of a shape normalization process is to make the shape recognition algorithms more robust to small...
shape variations caused by camera viewpoint orientation errors. The contour and region normalization were carried out by following the methods suggested in [79] and [80], respectively. For each test, the number of elements of the shape descriptor was: 64 (FD), 7 (BM), 8 (PCA), 12 (ZM) and 11 (CM). This was established after testing the methods for different numbers of elements and, eventually, choosing those which yielded the best shape recognition rates. In the case of the PCA descriptor, we used a set of 15 silhouettes for each viewpoint to compute the eigenvectors where there are six training images corresponding to viewpoints around the node, which signifies “moving” the camera slightly but always maintaining the model’s centroid in the optic axis. The other training images were obtained by adding Gaussian, salt and pepper and sparkle noise to the six training images.

**Sensor positions (W)**

We have carried out a test to evaluate the number of sensor positions needed to recognize an object. The experimental test was carried out on 14 objects in 10 different poses.

Figure 3.8 shows a set of comparative graphics of the accumulated recognition rate among different active strategies (Π, H, Δ) when using the shape descriptors mentioned previously (FD, HM, PCA, ZM, CM) for 14 sensor positions. As can be seen, when a single view is considered, the recognition rates are very low, but upon applying the active strategies, and after several sensor movements, the recognition rates exceed 90% in all cases.

Having analyzing the obtained results, we can confirm the importance of the shape recognition system in the overall process. The more efficient the shape recognition stage is the less uncertainty there is and, indeed, the fewer camera positions are necessary. Note that the active recognition systems based on boundary moments (HM) needs up to 14 camera positions to achieve a rate of 92%. This is owing to the high sensibility of the boundary moment method (HM) to shape variations. Nevertheless, for descriptors FD, PCA, ZM and CM, the active recognition systems are able, with few camera movements (five to seven images), to reach more than 94%.

We can conclude that, Complex Moments (CM) yielded the best results in the test. In spite of the fact that the Fourier descriptors (FD) use only the contour information of the object, which could be thought of as poor information, their performance is similar to that of the Zernike moments, which are region descriptors.

In order to evaluate the behavior of the different active strategy models, Figure 3.9 shows the recognition rates when we maintain the active strategy and test for each of the different shape descriptors. The heuristic (H) and possibilistic (Π) strategies yielded very similar results,
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

Figure 3.8: Recognition Rate in each step using different shape recognition methods. Each plot shows the active recognition rate for the selected descriptor with different active strategies (Possibilistic (Π), Heuristic(H), Deterministic(Δ)). a) Fourier Descriptors (FD). b) Boundary Moments (HM). c) Principals Components (PCA). d) Zernike Moments (ZM). e) Complex Moments (CM).

Figure 3.8: Recognition Rate in each step using different shape recognition methods. Each plot shows the active recognition rate for the selected descriptor with different active strategies (Possibilistic (Π), Heuristic(H), Deterministic(Δ)). a) Fourier Descriptors (FD). b) Boundary Moments (HM). c) Principals Components (PCA). d) Zernike Moments (ZM). e) Complex Moments (CM).

Note that the information in Figures 3.8 and 3.9 is the same, but plotted according to the variables in analysis (shape descriptors or active strategy).

Recognition cost of one iteration step (Σi)

We evaluated the average time taken by the shape recognition process (shape representation, shape identification and shape pose estimation) and the time needed to estimate the next best view position. The test was run on a 1.88 Ghz Pentium IV computer. Figure 3.10(a) illustrates the execution time for a single position camera when taking different combinations between active strategies and shape representation models. In this case, we considered 5 hypotheses. Note that the deterministic model has the highest computational cost. 3.10(b) shows a graph depicting, in seconds, the number of hypotheses versus the execution time for each active method (computing the evidence and the next sensor position). Remember that the number of hypotheses in our framework signifies the number of candidate views in the knowledge database. Of course, as the number of hypotheses increases, so does the time taken. This is exceeding rates of 80% for three captured images (FD, PCA, ZM, CM), whereas with the deterministic method, four images were necessary to achieve similar recognition percentages. Note that the HM shape recognition method clearly provided the worst results.

Note that the information in Figures 3.8 and 3.9 is the same, but plotted according to the variables in analysis (shape descriptors or active strategy).

Recognition cost of one iteration step (Σi)

We evaluated the average time taken by the shape recognition process (shape representation, shape identification and shape pose estimation) and the time needed to estimate the next best view position. The test was run on a 1.88 Ghz Pentium IV computer. Figure 3.10(a) illustrates the execution time for a single position camera when taking different combinations between active strategies and shape representation models. In this case, we considered 5 hypotheses. Note that the deterministic model has the highest computational cost. 3.10(b) shows a graph depicting, in seconds, the number of hypotheses versus the execution time for each active method (computing the evidence and the next sensor position). Remember that the number of hypotheses in our framework signifies the number of candidate views in the knowledge database. Of course, as the number of hypotheses increases, so does the time taken. This is
3.5 Experimentation

Figure 3.9: Recognition Rate in each step using different active strategy methods. Each plot shows the active recognition rate for the selected active strategy with different shape descriptors (FD, HM, PCA, ZM, CM). a) Posibilistic (Π). b) Heuristic (H). c) Deterministic (Δ)

mainly owing to the stage at which the D-Spheres corresponding to the candidates must be aligned. However, although this process is time consuming, the pose is obtained quickly. Observe that, for example, the system yields the pose in less than 0.1 seconds if ten candidates are considered.

Shape pose estimation error

Another aspect which must be taken into consideration is that of pose estimation accuracy. We have used the quadratic mean error \( E \) between the shape extracted from the image sample \( \vec{v}^{(k)} \) and the corresponding shape identified in the dataset \( \vec{v}^{\hat{I}}_{lo} \), and the pose parameter (rotation, translation scale) computed according to the representation model for Fourier Descriptors, Boundary Moments, Zernike Moments, Complex moments and PCA. Thus,

\[
E = \frac{\sum (\vec{v}^{(k)} - \vec{v}^{\hat{I}}_{lo})^2}{\text{ord}(\vec{v}^{\hat{I}})}
\] (3.36)

Table 3.2 shows the average of the quadratic mean error \( E \) and the execution times required to compute the pose for different shape representation models. Of all the methods, the Fourier Descriptors yield the best accuracy. Notice that the PCA descriptor presents the lowest time
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

![Figure 3.10: Recognition costs for the active recognition strategies. a) Time rates for a single position of the camera taking different combinations between active strategies and shape representation model ($\Sigma_z$). b) Time rates for the different active strategies when the number of hypotheses increases.](image)

rate but that its pose error is the highest.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error (pixels)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD</td>
<td>0.1377</td>
<td>0.023</td>
</tr>
<tr>
<td>HM</td>
<td>1.4671</td>
<td>0.029</td>
</tr>
<tr>
<td>PCA</td>
<td>2.9555</td>
<td>0.018</td>
</tr>
<tr>
<td>ZM</td>
<td>2.3306</td>
<td>0.086</td>
</tr>
<tr>
<td>CM</td>
<td>2.4374</td>
<td>0.179</td>
</tr>
</tbody>
</table>

Table 3.2: Average of the Quadratic Mean Error (Error) and execution time (Time) in pose estimation

**Recognition costs ($C$) and average of shape recognition rate ($R$)**

In Figure 3.11 it is possible to appreciate the performance of parameter recognition rate ($R_y$) and recognition costs ($C$). This graph allows us to consider the best combination between shape descriptors and active strategy. The best choice is the combination of maximal recognition costs (low time rate) and maximal recognition rate. Under this consideration, Fourier Descriptors and the possibilistic or heuristic active strategy seem to be the best choice. In general, region descriptors have a high recognition costs, despite requiring few robot movements.

**Average time taken to move from one position to the next ($\Sigma_m$)**

It is possible to decrease the recognition costs ($C$) by including a variable to weight the energy required to move the sensor from one position to another ($g_m$), into the cost function
3.5 Experimentation

![Figure 3.11: Active Recognition Performance. a) Recognition cost ($C$). b) Recognition Rate ($R$)](image)

In this way, the average time taken to move from one position to the next ($\bar{T}_m$) parameter could be reduced because the time consumption during sensor movement use to be higest than the time required to process the information at one sensor position ($\bar{T}_s$). Figure 3.12 shows the active recognition system computation cost but considering $g_m = 1$ in the cost function ($J$). Comparing the results from Figure 3.12 and Figure 3.11(a) we can see that the recognition costs decreases between 17 to 25 seconds.

![Figure 3.12: Computational efficiency ($\bar{C}(s)$) considering $g_m = 1$](image)

**Results discussion**

It is possible to conclude from the results obtained in Experiments 1 that:

- After several robot positions, no matter what kind of descriptor is used, the recognition rates rises by 90%. Parameter $R$ affects the number of sensor positions necessary to
achieve high recognition rates. Boundary Moments (HM) is a good example of this problem. Its low recognition rate implies a large number of robot movements.

- The shape pose error becomes a "measure" of the expected error of the object pose and the applicability of the implemented active recognition system to satisfy the accuracy constraints in the task to be developed. Moreover, during the selection of the next sensor position, the computed shape pose parameter is used. What is more, huge errors in the rotation parameter could lead to errors in the estimation of the next sensor position, thus increasing the number of sensor positions required. It is for this reason that PCA shows a worse performance than Fourier Descriptors under the same conditions.

- Upon considering recognition rates versus recognition costs, the implementation of our framework using FD and Heuristic strategy provides the best results. It is important to realize that descriptors based on shape region must be used in robotic systems with high processing capabilities because of their high computational costs. The combination of CM and the Heuristic strategy is the best choice in these cases.

- Active strategies based on heuristic and possibilistic methodologies show a better performance than deterministic methodologies.

3.5.3 Experimentation 2

The objective of this subsection is to compare the performance of our proposed framework with other active recognition frameworks. The comparative analysis has been developed by implementing another two representative frameworks in the representation of the main active framework methodologies: stochastic and deterministic. The selected frameworks are: Borotschnig et al. [82] and Kovacic et al. [89]. The former is based on stochastic methods, whereas the latter uses deterministic models.

In order to make a comparative analysis, our framework has been implemented by means of the methods shown in Table 3.3.

In Borotschnig et al. [82] three different strategies are compared using stochastic models. The results from this comparative study show that the probabilistic model is the best choice. We have also implemented this probabilistic model. In our case, a set of training views and the representation of the information of this views by means of PCA [96] is required.
3.5 Experimentation

<table>
<thead>
<tr>
<th>( \vec{v} )</th>
<th>shape silhouette</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \vec{w} = \Phi(\vec{v}) )</td>
<td>( \vec{w} = FD ) [90]</td>
</tr>
<tr>
<td>( D(v_i, v^{(k)}) )</td>
<td>( \sqrt{\Sigma(w_l, w^{(k)})^2} )</td>
</tr>
<tr>
<td>( D({v_i}) )</td>
<td>( \text{min}D(\vec{v}<em>{m,t}, \vec{v}</em>{m,p}), 1 \leq t, p \leq q, t \neq p, q ) is the number of hypothesis</td>
</tr>
<tr>
<td>( E_j^{(k)} )</td>
<td>( H E_j^{(k)} )</td>
</tr>
<tr>
<td>( \rho_m )</td>
<td>( \vec{H} \vec{A} )</td>
</tr>
<tr>
<td>( B^k )</td>
<td>( \text{arg}_{l}D(w_l, w^{(k)}) &lt; \varepsilon )</td>
</tr>
<tr>
<td>( P(w_l, w^{(k)}) )</td>
<td>if ( \vec{w} = FD ) [90]</td>
</tr>
<tr>
<td>( g_m = \Sigma(\Theta^{(k)}, \Theta_m) )</td>
<td>( \sqrt{\Sigma(\Theta^{(k)}, \Theta_m)}^2 ) where ( \Theta ) are the cartesian coordinates of ( \Theta ) robot join variables</td>
</tr>
<tr>
<td>( \varsigma_m )</td>
<td>Configured off-line considering whether the node ( m ) can be accessed by the robot end-effector</td>
</tr>
<tr>
<td>( \mathcal{J} )</td>
<td>( \text{max}_m (\rho_m : g_m : \varsigma_m) )</td>
</tr>
</tbody>
</table>

Table 3.3: Collection of methods used to implement the proposed framework to develop a comparative analysis

The second method selected is that developed by Kovacic et al. in [89]. This framework is based on an off-line clustering process. The off-line process provides the active system with high recognition costs. In this case, we have used the FD to represent the shapes and the clustering process from [90], and the framework from [89].

From here on, our framework, the system of Borotschnig et al. ([82]) and that of Kovacic et al. in ([89]) will be denoted as AR 1, AR 2 and AR 3.

The comparative analysis takes into consideration the performance of AR 1, AR 2 and AR 3 in a robotic platform. We have also evaluated their behavior in a simulator in order to collect information about the object pose accuracy.

3.5.3.1 Robotic Platform

All three active recognition systems have been compared using the same parameters used in Experiment 1: Recognition rate (\( R \)) and Recognition cost (\( C \)). Since the behavior of the active recognition is dataset dependent, we have developed the tests using Dataset 1, Dataset 2 and Dataset 3. This comparison allows us to discover the sensibility of each method to different datasets.

We selected 6 objects, three from Dataset 2 and three from Dataset 3, and ran the active recognition systems (AR 1, AR 2, AR 3) using Dataset 1 and the dataset corresponding with
the scene object. A total of 60 tests were carried out (10 tests per object). The test results are shown in Figure 3.13.

![Figure 3.13: Active recognition systems performance for Dataset 1, Dataset 2 and Dataset 3. a) Recognition Rate (\(R\))(%) b) Recognition cost (\(C\))(s).](image)

From Figure 3.13 we can conclude that the AR 3 framework has a low recognition rate, although its recognition cost is the lowest. The AR 1 and AR 2 frameworks have the same recognition rate for Dataset 3 but AR 1 improves on AR 2 in the case of a dataset with non-ambiguous objects (Dataset 2). AR 1 shows the best performance for Dataset 1. Upon comparing the recognition cost parameter between AR 1 and AR 2, AR 1 has the lowest value for Dataset 1 and Dataset 2, in case of Dataset 3, the recognition cost of AR1 is slightly higher than AR2.

Moreover, the recognition rate differences between AR 3 versus AR 1 and AR 2 suggest that, using explicit sensor planning, the computation cost is low but the recognition rate show a low performance.

### 3.5.3.2 Recognition Simulator

The experiments developed in the robotic platform do not allow a quantitative measure of the object pose estimation accuracy to be obtained. This parameter can be evaluated by using an
3.5 Experimentation

object recognition simulator. A set of tests were carried out in a simulated framework in order to evaluate the pose estimation parameter ($P_o$) for AR 1, AR 2 and AR 3.

The recognition simulator has been developed using, samples of synthetic views. These synthetic views correspond with the object models from the dataset after random transformations. Moreover, a random rotation in axes X and Y is applied to the cloud of points of object $o_i$ in order to achieve the query object $\hat{d}_i$. The simulator therefore uses the projection of $\hat{d}_i$ from the viewpoint corresponding to the current sensor position as a sample view. The pose accuracy parameter ($P_o$) is computed by means of the mean quadratic error between query object ($\hat{d}_i$) and the ($o_i$) object after being transformed by the estimated pose parameters ($\hat{\hat{d}}_i$) using the next equation. See [100] for more information about the rotation process.

Let $\hat{x}_i, \hat{y}_i, \hat{z}_i$ be the coordinates associated with the cloud of points for the sample ($\hat{d}_i$) and $\hat{x}_i, \hat{y}_i, \hat{z}_i$ be the coordinates of the candidate after applying a transformation $\hat{T}$ estimated by the active system. The pose error is computed by:

$$P_o = \sqrt{\frac{\sum (\hat{x}_i - \hat{\hat{x}}_i)^2 + (\hat{y}_i - \hat{\hat{y}}_i)^2 + (\hat{z}_i - \hat{\hat{z}}_i)^2}{\hat{n}}}$$

where $\hat{n}$ is the number of points in the sample.

Table 3.4 shows the recognition cost ($C_o$), recognition rate ($R$), and pose accuracy parameter ($P_o$) in the simulator for 54 randomly rotated objects.

<table>
<thead>
<tr>
<th>Active model</th>
<th>$C_o$</th>
<th>$R$</th>
<th>$P_o$ (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AR 1)</td>
<td>0.23</td>
<td>97.0</td>
<td>0.023</td>
</tr>
<tr>
<td>(AR 2)</td>
<td>0.30</td>
<td>95.3</td>
<td>0.61</td>
</tr>
<tr>
<td>(AR 3)</td>
<td>0.10</td>
<td>92.9</td>
<td>0.017</td>
</tr>
</tbody>
</table>

Table 3.4: Comparison between different active recognition systems in the active simulator

The results in Table 3.4 show that AR 1 performs better than the other two active recognition systems. Upon comparing the recognition rate parameter ($R$) of AR 3 in the simulation and on the robotic platform, we can see that the difference is 15%, while the averages for AR 1 and AR 2 are very similar on both platforms. This is an indicator of the low robustness of AR 3 to scene variations (noise, small viewpoint variations, etc).

The pose estimation parameter ($P_o$) of AR 2 has the highest error. In case of AR 2, it is necessary to increase the number of views in the dataset in order to increase the pose estimation accuracy, meanwhile, computing the object pose using the geometric properties of the shape.
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR ACTIVE RECOGNITION SYSTEMS

descriptor (AR 1), the pose estimation parameter show better performance than the probabilistic system. Figure 3.14 show several examples of the pose estimation results for each active system.

Figure 3.14: Examples of pose estimation using different active recognition systems. (a) AR 1. (b) AR 2. (c) AR 3.

3.6 Conclusions

In this chapter we have defined a complete framework with which to permit the implementation of active recognition systems based on monocular vision. This framework is also independent
of the type of shape descriptors (contour or region based), and the similarity measure used to
develop the shape recognition process. It is mainly focused on solving the uncertainty problem
that is present in appearance-based object recognition systems by moving the robot to positions
at which the sensor measurements allow a rapid discrimination of the object hypothesis. This
framework is robust to scene variations, and maintains an historical register of view evidence
at each sensor position. In order to allow the use of non stochastic shape descriptors, we have
proposed three active strategies based on possibilistic, heuristic and deterministic methodolo-
gies. The framework was implemented by using different shape descriptors, and the Euclidean
distance as a similarity measure. A comparative analysis was therefore developed in order to
study the framework’s performance for: (i) different combinations of the methods involved
in active recognition stages, and (ii) datasets with different characteristics. We have addition-
ally implemented another two well known frameworks, and they have been compared with
our framework with regard to recognition rates and recognition costs parameters. Experiments
showed that our framework exhibits a better performance in these two indexes. Even more,
the experiments proves that several handicaps of the active recognition systems from refer-
ences are overcome. Our framework has been experimentally tested with a 6 DOF manipulator
robot, and has shown promising results.
3. MULTIPLE VIEW-BASED 3D OBJECT RECOGNITION. A FRAMEWORK FOR
ACTIVE RECOGNITION SYSTEMS
This chapter presents an active recognition system (CSS-NSSD system) that will allow object recognition to be developed in large multi-view object datasets. The active recognition system is addressed towards simplifying the D-sphere structure in order to make the recognition active method simpler and faster. Most of the ambiguity problems are owing to symmetries in the object model. Using all the nodes of the tessellated sphere therefore implies storing redundant views and information in the representation model. The proposed object recognition system is focused on satisfying the main constraints of any recognition system: high computational efficiency, high recognition rate and high accuracy for object pose estimation. The good performance of the proposed object recognition system is based on an efficient and normalized object representation model.

4.1 Introduction

Several active recognition methodologies have been developed over the years [82, 85, 89, 100]. These recognition systems show high recognition rates even in the case of objects with a high level of ambiguity, but their computational efficiency is low in large datasets. A large number of objects and views must be handled in the dataset, and many robot movements are required to identify an object in the scene. These methods are usually focused on solving the ambiguity/uncertainty problem without considering the object representation model. Furthermore, a large amount of views representing the object model increases the number of items
4. ACTIVE OBJECT RECOGNITION BASED ON CSS-NSSD

in the dataset, and the searching process to identify the scene in the dataset therefore become slower. In most cases researchers use a probabilistic framework to solve the uncertainty problem [82, 85], which requires a shape descriptor with stochastic properties, such as the PCA descriptor [98]. However, active recognition systems based on PCA have an important drawback: the pose of the object cannot be accurately estimated, signifying that object manipulation tasks are unviable. It is therefore advisable to choose other shape representation models to provide accuracy in object pose estimation. Although shape representation models are a widely researched issue [101], their implementation in active object recognition systems implies several constraints such as: robustness to shape variations caused by viewpoint variation, noise, illumination changes, etc. Researchers deal with these problems by using a feature vector that is invariant to affine transformations, but the critical problem concerns the number of elements in the feature vector that are necessary to describe, with a given precision, all the shapes in a database in which the number of different shapes is very high. From our experience it can be stated that the recognition rates increase or decrease according to the number of elements in the feature vector and that, depending on the shape boundary, more or less elements must be used to describe it. Moreover, if a new object is added to the database, most recognition systems have to re-run several tests in order to find the optimal number of elements in the feature vector.

The contribution of our work lies on defining a database object representation model in an attempt to solve the two questions which will be dealt with in the following sections: (1) the number of views that are necessary to model an object, and (2) the shape representation model which is invariant to affine transformations and is able to represent all the shapes in the dataset with the same precision. We first propose a method with which to select the so-called Canonical Sphere Section (CSS) by taking into account the symmetry properties in the object. Each canonical view corresponding to its CSS is then characterized in the Fourier spectral domain. This representation is denominated as the Normalized Silhouette in the Fourier Spectral Domain (NSSD) model, it is invariant to affine transformations and is able to represent all the dataset shapes with the same precision using a feature vector of a fixed number of elements. This proposed database representation model has been implemented in an active recognition system and compared with others. The following sections provide a detailed explanation of the database object representation model and its performance in a robotic-vision system.
4.2 A simplified Representation Model

4.2.1 Defining the Canonical Spherical Section (CSS)

Traditional approaches to symmetry detection work with discrete symmetries under rotation, reflection, or translation. For example, Zank and Huebner [102] describe efficient algorithms which extract the whole-object symmetries by using substring matching. They use an octree representation (developed by Minovic et al. [103]) and the singular value decomposition of the points of the model (Shah and Sorensen [104]). However, these methods are sometimes inefficient when the number of views and the homogeneity of the views are not suitable. The approaches developed by Thrun and Wegbreit [105] and Podolak et al. [106], contain more stable and efficient models, and are based on the detection of rotations and reflective symmetries. These methods are more stable but considerably increase the computational complexity in the symmetry detection phase. The efficiency of these methods demonstrates that the reflective symmetry is a simple property which can be implemented in canonical spheres. In this section we describe how to use the reflective symmetry property to characterize objects using a canonical sphere. The idea of reflective symmetry intuitively means that one or more than one symmetry planes can be defined in the object so that the portions within these planes can be viewed by an observer in an equal (or reflected) manner. In order to integrate this idea into a canonical sphere, we associate each repeated portion (equal or reflected) of the object with a portion of the canonical sphere as follows.

The sphere coordinate system is first aligned with the object coordinate system, which has been established according the principal directions of the object. The principal directions of the object can be defined by using a PCA process (Principal Component Analysis) on the 3D data of the object. More specifically, axis Y is aligned with the principal axes of the object. Figure 4.1(a) shows a Canonical Sphere $S$ represented in the canonical coordinate system, Figure 4.1(b) shows the principal axis of the object and Figure 4.1(c) shows the result of the alignment of the principal object axes with the canonical coordinates system.

After the object and the canonical sphere have been aligned, we consider the reflective symmetry planes which contain axes $Z$ and $Y$. The number of symmetry planes passed through by $Z$ and $Y$ therefore defines the so-called reflective symmetry orders $n_z$ and $n_y$. The reflective symmetry of the object is thus synthesized by the pair $(n_z,n_y)$. Since a portion of the object is repeated or reflected in the object, its representation model can be reduced to only one representative portion in the canonical sphere. From here on we shall call this portion the Canonical
4. ACTIVE OBJECT RECOGNITION BASED ON CSS-NSSD

Figure 4.1: a) Sphere coordinates system b) Object principal axes. c) Object principal axis aligned with the reference coordinate system.

Sphere Section (CSS). Figure 4.2 shows the CSSs corresponding to a collection of objects which have reflective symmetry properties. Note the symmetry orders \( n_z \) and \( n_y \) for each case. The CSS concept therefore intuitively leads us to a reduced view-base representation database. A formal CSS definition is introduced in the following paragraphs.

Figure 4.2: Samples of objects with different symmetries, their Canonical Object Section and their Canonical Sphere Section.(a) \( n_y = 1 \) and \( n_z = 0 \). (b) \( n_y = 2 \) and \( n_z = 0 \). (c) \( n_y = 4 \) and \( n_z = 0 \). (d) \( n_y = 20 \) and \( n_z = 2 \).

4.2.2 Defining the Canonical Spherical Section (CSS)

Since the Canonical Spherical Section (CSS) is related to the reflective symmetry order for an object, the first step is to define the relationship between the reflective symmetry order and the Canonical Sphere Section.

Let us assume an object model, \( o \), whose principal axes have been aligned with a canonical coordinate system, as in Figure 4.1. The model \( o \) is viewed from different view directions
4.2 A simplified Representation Model

corresponding to the nodes of the sphere $S$ with $J$ nodes and the image is projected in the
direction defined by the node with the camera pointing towards the sphere center. Let $I_j, 1 \leq j \leq J$ be the synthetic image of the object from node $j$. The position of node $j$, which is
representative of the view $j$, is given in spherical coordinates by a pair of angles $(\psi_j, \sigma_j), 
\psi_j \in [-\pi, \pi]$ and $\sigma_j \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, where $\psi_j, \sigma_j$ are the azimuth and elevation coordinates.

Let $n_y$ and $n_z$ be the reflective symmetry orders, $\Xi^*$ be the Canonical Sphere Section, and $\Xi$ be the rest of the canonical sphere. We shall denote $I_j^*, j^* \in \Xi^*$ the synthetic images correspond-
ing to the Canonical Sphere Section and $I_j, j \in \Xi$ the synthetic images corresponding to
the rest of the canonical sphere. Note that if $n_y \neq 0$, the nodes of $\Xi^*$ verifying $\psi_j \in [0, \frac{\pi}{n_y}]$, and in the case of $n_z \neq 0$ the nodes verifying $\sigma_j \in [0, \frac{\pi}{n_z}]$.

The images of the object from the nodes belonging to $\Xi$ must be equal to or reflected by
images from the nodes of $\Xi^*$. One of the following equations must therefore be verified:

$$
if \quad \psi_j = (k_y \cdot \frac{2\pi}{n_y} + \psi_j^*), \quad \sigma_j = (k_z \cdot \frac{\pi}{n_z} + \sigma_j^*) \quad then \quad I_j = I_j^* \quad (4.1)
$$
or

$$
if \quad \psi_j = (k_y \cdot \frac{2\pi}{n_y} - \psi_j^*), \quad \sigma_j = (k_z \cdot \frac{\pi}{n_z} - \sigma_j^*) \quad then \quad I_j = \hat{I}_j^* \quad (4.2)
$$

where $\hat{I}_j^*$ is the reflected image of $I_j^*, k_y \leq 2n_y, k_z \leq 2n_z, k_y, k_z \in \mathbb{N}$.

Figure 4.3 shows an example of the relationship between one view (image) from the Canonical
Spherical Section and its corresponding images (reflected or not) in the rest of the canonical
sphere.

Note that if at least one symmetry plane is in the plane $XY$ then $n_y > 0$ and in the case of
$n_z > 0$, one symmetry plane must be in the plane $XZ$.

4.2.3 Computing the Canonical Spherical Section (CSS)

Based on equations (4.1) and (4.2), it is simple to compute the order of the reflective symmetry
$n_y$ and $n_z$ over axes $Y$ and $Z$ respectively. Algorithms 1 and 2 show the process used to compute
$n_y$ and $n_z$ values.

Figure 4.4 illustrates the views corresponding to the Canonical Spherical Section (CSS) for
an object model.
Algorithm 1 Computing $n_y$

Require: $\sigma^*$ coordinate

Require: $\psi^*$ coordinate

1: $\hat{n}_y = 0$
2: while flag do
3:  $\hat{n}_y = \hat{n}_y + 1$
4:  for $k_y = 1 : \hat{n}_y$ do
5:  if $k_y$ is odd then
6:    $\psi_r = (k_y \cdot \frac{2\pi}{\hat{n}_y}) - \psi^*$
7:  else
8:    $\psi_e = (k_y \cdot \frac{2\pi}{\hat{n}_y}) + \psi^*$
9:  end if
10: isReflection=CompareReflected($I(\psi_r, \sigma^*), I(\psi^*, \sigma^*)$)
11: isEqual=CompareEqual($I(\psi_e, \sigma^*), I(\psi^*, \sigma^*)$)
12:  if (not isReflection) or (not isEqual) then
13:    flag = not flag
14:  BREAK
15: end if
16: end for
17: end while
18: $n_y = \hat{n}_y - 1$
4.2 A simplified Representation Model

Algorithm 2 Computing \( n_z \) order

Require: \( \sigma^* \) coordinate

Require: \( \psi^* \) coordinate

1: \( \hat{n}_z = 0 \)

2: while flag do

3: \( \hat{n}_z = \hat{n}_z + 1 \)

4: for \( k_z = 1 : \hat{n}_z \) do

5: if \( k_y \) is odd then

6: \( \sigma_r = (k_z \cdot \frac{\pi}{\hat{n}_z}) - \sigma^* \)

7: else

8: \( \sigma_e = (k_z \cdot \frac{\pi}{\hat{n}_z}) + \sigma^* \)

9: end if

10: isReflection=CompareReflected(\( I(\psi^*, \sigma_r), I(\psi^*, \sigma^*) \))

11: isEqual=CompareEqual(\( I(\psi^*, \sigma_e), I(\psi^*, \sigma^*) \))

12: if (not isReflection) or (not isEqual) then

13: flag = not flag

14: BREAK

15: end if

16: end for

17: end while

18: \( n_z = \hat{n}_z - 1 \)
4. ACTIVE OBJECT RECOGNITION BASED ON CSS-NSSD

Figure 4.3: Example of the relationship between one view (image) from the Canonical Spherical Section and its corresponding views in the canonical sphere.

4.3 NSSD descriptor

The most popular methods for 2D object recognition from silhouettes are based on invariant moments or Fourier descriptors. Invariant moments exhibit the drawback that two completely different silhouettes might have the same low order invariant moments, which may lead to ambiguities in the recognition process. Fourier descriptors yield much more information about the silhouette, and only similar silhouettes exhibit similar Fourier descriptors. Shapes represented in the Fourier spectral domain are invariant to rotation, translation and scale. It is also possible to represent, with high precision, a silhouette using a low number of Fourier descriptors, and these can be efficiently used for 3D object recognition from 2D images [90]. However, Fourier descriptors are sensitive to shape variations and variant to affine transformations. In order to achieve a shape descriptor that is invariant to affine transformation and more robust to shape variations than the traditional Fourier descriptor, we propose a shape representation model in the Fourier domain (NSSD descriptor).

The NSSD descriptor is developed in the Fourier framework by means of three process:

1. Contour normalization

2. Precision normalization

3. Reduction of Fourier Descriptors
The contour normalization process is equivalent to that proposed in [79] but is reformulated in the spectral domain. This leads to several notable improvements. Firstly, the object contour is normalized and become invariant to rotation, translation, scale and skew. Moreover, the robustness of the method increases owing to the use of the spectral domain. Nevertheless in [79] the normalization process to the starting point (rotation) is not ensured, whereas rotation invariance in the spectral domain is guaranteed.

The reduction of the Fourier descriptors and the precision normalization processes are carried out by bearing in mind that the most significant harmonics lie in the first and last positions of the Fourier descriptor vector, whereas unimportant contour details are contained in the central harmonics.

4.3.1 Affine Normalization

Assume the projected image of an object \( o \) from a generic viewpoint and assume its silhouette \( s \) composed of \( N \) points \( s(n) = (x(n), y(n)) \), \( 1 \leq n \leq N \) on the plane \( XY \) where the origin of index \( n \) is an arbitrary point of \( s \), and \( n \) and \( n + 1 \) are consecutive points according to a given direction (for example, a clockwise direction) on the silhouette. In order to normalize the silhouette \( s \) to affine variations (scale, rotation, translation and skew), a set of linear functions are applied to the silhouette spectral representation based on the contour orthogonalization (Avrithis et al. [79]). The Fourier Transform \( (\mathcal{F}) \) is thus applied to the silhouette \( s(n) \), achieving \( S(m) = X(m) + iY(m) \), where \( X(m) = \mathcal{F}(x(n)) \) and \( Y(m) = \mathcal{F}(y(n)) \), where \( 1 \leq m \leq M, M = \ldots \)
4. ACTIVE OBJECT RECOGNITION BASED ON CSS-NSSD

N. Thus for each silhouette transformation in the spatial domain in (Avrithis et al. [79]), an equivalent operation is developed in the Fourier spectral domain. Note, that in each step, we made reference to the step developed in the spatial domain by [79] and then it is formulated the equivalent transformation that must be applied to the Fourier descriptors, so that if at each step we apply the inverse Fourier transform to the achieved Fourier descriptors, the resulting silhouette is the same as the achieved by (Avrithis et al. [79]).

1. The center-of-mass of the silhouette is normalized so as to coincide with the origin ($S \rightarrow S_1$):

\[ X_1(1) = 0, X_1(m) = X(m), 2 \leq m \leq M \]
\[ Y_1(1) = 0, Y_1(m) = Y(m), 2 \leq m \leq M \] (4.3)

Note that $\mu_x = X(1), \mu_y = Y(1)$ are the displacement of center-of-mass.

2. The silhouette is scaled horizontally and vertically ($S_1 \rightarrow S_2$):

\[ X_2(m) = X_1(m) \cdot \rho_x, \quad Y_2(m) = Y_1(m) \cdot \rho_y \] (4.4)

where $\rho_x = \frac{1}{\sqrt{\sum (X_1(m))^2}}, \rho_y = \frac{1}{\sqrt{\sum (Y_1(m))^2}}$

3. The silhouette is rotated $\pi/4$ counterclockwise ($S_2 \rightarrow S_3$):

\[ X_3(m) = \frac{N}{\sqrt{2}} \cdot (X_2(m) - Y_2(m)), \quad Y_3(m) = \frac{N}{\sqrt{2}} \cdot (X_2(m) + Y_2(m)) \] (4.5)

4. The silhouette is horizontally and vertically scaled ($S_3 \rightarrow S_4$):

\[ X_4(m) = X_3(m) \cdot \tau_x, \quad Y_4(m) = Y_3(m) \cdot \tau_y \] (4.6)

where $\tau_x = \frac{N}{\sqrt{\sum (X_3(m))^2}}, \tau_y = \frac{N}{\sqrt{\sum (Y_3(m))^2}}$

Finally, the silhouette feature vector is:

\[ S_4(m) = X_4(m) + i \cdot Y_4(m) \] (4.7)

where $|S_4(m)|$ is invariant to rotation, translation, scale and skew transformations.

In order to allow the pose estimation between two silhouettes, it is necessary to store in the dataset $\mu_x, \mu_y, \rho_x, \rho_y, \tau_x, \tau_y$ and $N$ corresponding to each silhouette.
4.4 NSSD descriptor properties

4.3.2 Precision Normalization and Dimensionality Reduction

The precision normalization of descriptors is based on replacing the central harmonics by zeros. Then the descriptors vector would have only \( K \) significant values, being \( K < M \) an even number.

If \( S_4 = \{ S_4(1), \ldots, S_4(M) \} \), the normalized vector would be of the form

\[
S_4^{(K)} = \{ S_4(1), S_4(2), \ldots, S_4(K/2), 0, \ldots, S_4(M + 1 - K/2), \ldots, S_4(M) \} \tag{4.8}
\]

The precision normalization stage is carried out in such way that the spatial representation error between the original and the normalized silhouettes is smaller than a predefined precision error \( \mathcal{E} \). In this sense, the precision normalized vector is \( S_4^{(K^o)} \) where

\[
K^o = \min K, K \in \mathbb{N} | |F^{-1}(S_4) - F^{-1}(S_4^{(K)})| \leq \mathcal{E} \tag{4.9}
\]

being \( | \cdot | \) the Euclidean norm.

The reduction of the number of descriptors is based on eliminating the central harmonics achieving a new vector with the desirable length \( \mathcal{H} \), where \( \mathcal{H} < M \). Thus, if we have the \( S_4^{(K^o)} \) defined in (4.8), the final descriptor \( \hat{S}_4^{(K^o)} \), which is called NSSD (Normalized Silhouette in the Spectral Domain) descriptor, is defined as follow:

\[
\hat{S}_4^{(K^o)} = \{ S_4(1), S_4(2), \ldots, S_4(\mathcal{H}/2), S_4(M + 1 - \mathcal{H}/2), \ldots, S_4(M) \} \tag{4.10}
\]

4.4 NSSD descriptor properties

The performance of our NSSD descriptor can be evaluated by means of three properties: (1) invariance to parameters such as translation, scale, rotation, skew, etc. (2) stability and (3) completeness. A stable set of invariants guarantees that small shape variations do not induce a noticeable change in the features [34]. This property provides robustness under small distortions of images. On the other hand, a complete set of invariants also guarantees that an object is fully identified under any transformation [107]. Finally, completeness is an important criterion for full shape discrimination and feature reconstruction [108]. The following paragraphs illustrate how NSSD verifies these properties.

**Invariance**
To demonstrate the invariance, figure 4.5(a) shows an example in which several affine transformations have been applied on a particular silhouette. Figure 4.5(b) presents the NSSD computed for all these silhouettes. Note that NSSD descriptors have equal or very close values. This result confirm the invariance of this set of descriptors under affine transformations.

![Figure 4.5](image-url)

**Figure 4.5**: Example of NSSD descriptor invariance property. (a) Samples of affine transformations for the original Silhouette. (b) Superposition of NSSD descriptors (module) for different affine transformations.

**Stability**

In order to test the stability, a small distortion is made on the original silhouette and the respective NSSD descriptors are compared. An example of original and affine distorted silhouettes appears in Figures 4.6(a) and 4.6(b). Figure 4.6(c) depicts the module of NSSD descriptor for both silhouettes and Figure 4.6(d) shows the module of traditional Fourier Descriptors [39]. Comparing errors in figure 4.6(c) and 4.6(d), we can see that the NSSD descriptors are more stable than traditional FD’s ones.

**Completeness**

The completeness property signifies that after applying an affine transformation on a silhouette, it is possible to recover the original shape from the distorted silhouette.

Assume $s$ a original silhouette represented by its NSSD descriptor $\hat{S}^{(K)}_4$, obtained using the process described in Section 4.3.2. Let $(\tau_x, \tau_y, \rho_x, \rho_y, \mu_x, \mu_y)$ be the parameters used to normalize $s$ (to obtain $S_4$), and $M$ be the length of $s$ and of $S = \mathcal{F}(s)$. To reconstruct $s$ from $\hat{S}^{(K)}_4$, the operations presented in Section 4.3.1 are applied in inverse order:

1. The silhouette is scaled

$$
\hat{X}^{(K)}_3(\bar{h}) = \frac{\hat{X}^{(K)}_4(\bar{h})}{\tau_x}, \quad \hat{Y}^{(K)}_3(\bar{h}) = \frac{\hat{Y}^{(K)}_4(\bar{h})}{\tau_y}, \quad 1 \leq \bar{h} \leq \mathcal{H} \quad (4.11)
$$
4.4 NSSD descriptor properties

Figure 4.6: Example of stability property. (a) The original silhouette. (b) Distorted silhouette. (c) NSSD descriptor module for the original and the distorted silhouette. The mean square error is 0.0014852. (d) FD descriptor module for the original and the distorted silhouette. The mean square error is 0.0030151

2. The silhouette is rotated $-\pi/4$ rads counterclockwise

$$
\hat{X}_2^{(K^\nu)}(\bar{h}) = \frac{1}{\sqrt{2}}(\hat{X}_3^{(K^\nu)}(\bar{h}) + \hat{Y}_3^{(K^\nu)}(\bar{h})), \quad \hat{Y}_2^{(K^\nu)}(\bar{h}) = \frac{1}{\sqrt{2}}(\hat{Y}_3^{(K^\nu)}(\bar{h}) - \hat{X}_3^{(K^\nu)}(\bar{h})), \quad 1 \leq \bar{h} \leq H
$$

(4.12)

3. The silhouette is scaled horizontally and vertically

$$
\hat{X}_1^{(K^\nu)}(\bar{h}) = \frac{\hat{X}_2^{(K^\nu)}(\bar{h})}{\rho'_x}, \quad \hat{Y}_3^{(K^\nu)}(\bar{h}) = \frac{\hat{Y}_2^{(K^\nu)}(\bar{h})}{\rho'_y}
$$

(4.13)

4. The centroid of the silhouette is normalized so as to coincide with the origin. Formally

$$
\hat{X}^{(K^\nu)}(\bar{h}) = \hat{X}_1^{(K^\nu)}(\bar{h}), \quad 2 \leq \bar{h} \leq H; \quad \hat{X}^{(K^\nu)}(1) = \mu'_x \cdot M
$$

$$
\hat{Y}^{(K^\nu)}(\bar{h}) = \hat{Y}_1^{(K^\nu)}(\bar{h}), \quad 2 \leq \bar{h} \leq H; \quad \hat{Y}^{(K^\nu)}(0) = \mu'_y \cdot M
$$
4. ACTIVE OBJECT RECOGNITION BASED ON CSS-NSSD

Then, $\hat{S}^{(K)}(\bar{h}) = \hat{X}^{(K)}(\bar{h}) + i \cdot \hat{Y}^{(K)}(\bar{h}), \quad 1 \leq \bar{h} \leq \mathcal{H}$

Moreover, $\hat{S}^{(K)}$ must be modified adding $M - \mathcal{H}$ zeros to achieve the length of the original silhouette:

$$S^{(K)} = \{\hat{S}^{(K)}(1), \hat{S}^{(K)}(2), \hat{S}^{(K)}(\mathcal{H}/2), 0 \ldots 0, \hat{S}^{(K)}(\mathcal{H}/2 + 1) \ldots \hat{S}^{(K)}(\mathcal{H})\} \quad (4.14)$$

and $s^{(K)} = \mathcal{F}^{-1}(S^{(K)})$, being $s^{(K)}$ the spatial image of the NSSD descriptors $\hat{S}^{(K)}$. Then the completeness property means that $s \approx s^{(K)} \cdot e^{i\phi}$ where $\phi$ is the rotation angle between silhouettes. The estimation of $\phi$ is presented in the next section.

Figure 4.7 shows an example of completeness.

![Figure 4.7: Example of completeness. (a) The original silhouette and distorted silhouette after applying an affine transformation. (b) Reconstructed silhouette from invariants NSSD.](image)

4.5 Estimation of pose parameters

Assume two silhouettes $s$ and $\bar{s}$ represented by the NSSD descriptor $S^{(K)}_4$ and $\bar{S}^{(K)}_4$ respectively. The pose parameters are defined by rotation $(\phi)$, translation $(\vec{\gamma})$ and scale $(\vec{\rho})$ of $\bar{s}$ with respect to $s$. Let $(\mu_x, \mu_y), (\bar{\mu}_x, \bar{\mu}_y)$ be the centroids and $(\rho_x, \rho_y), (\bar{\rho}_x, \bar{\rho}_y)$ the scale factor for $S^{(K)}_4$ and $\bar{S}^{(K)}_4$, computed following the procedure of Section 4.3.2. The translation $(\vec{\gamma})$ and the scale $(\vec{\rho})$ are estimated as follows:

$$\vec{\gamma} = (\bar{\mu}_x - \mu_x, \bar{\mu}_y - \mu_y) \quad (4.15)$$

$$\vec{\rho} = \left(\frac{\bar{\rho}_x}{\rho_x}, \frac{\bar{\rho}_y}{\rho_y}\right) \quad (4.16)$$
4.5 Estimation of pose parameters

Firstly note that silhouette $s^{(K)}$ (the image of the NSSD descriptors $\hat{S}_4^{(K)}$) coincides with the silhouette $\tilde{s}^{(K)}$ (the image of the NSSD descriptors $\hat{S}_4^{(K)}$) rotated ($\phi$) degrees. Then we have that:

$$s^{(K)}(\tilde{h}) = D(s^{(K)}, \delta) \exp(j\phi), 1 \leq \tilde{h} \leq \mathcal{H}$$

(4.17)

where $D(s^{(K)}, \delta)$ displaces $\delta$ units the origin of the sequence of $s^{(K)}(\tilde{h})$. Its Fourier transform is:

$$\tilde{S}^{(K)}(\tilde{h}) = e^{j\phi} e^{j2\pi\delta\tilde{h}/M} S(\tilde{h})$$

(4.18)

If the cost function:

$$f(q, \delta) = (\tilde{S}^{(K)} - qP(\delta))^* (\tilde{S}^{(K)} - qP(\delta))$$

(4.19)

where $q = e^{j\phi}$, $\tilde{S}^{(K)} = (\tilde{S}^{(K)}(1), ..., \tilde{S}^{(K)}(\mathcal{H}))^T$, $P(\delta) = (S^{(K)}(1), S^{(K)}(2)e^{j2\pi\delta/2\mathcal{H}}, ..., S^{(K)}(\mathcal{H} - 1)e^{j2\pi\delta(\mathcal{H} - 2)/2\mathcal{H}}, S^{(K)}(M)e^{j2\pi\delta(\mathcal{H} - 1)/M})$, * denotes conjugate and $t^*$ denotes transpose conjugate, is minimized with respect to the complex number $q$, it yields

$$q^0(\delta) = \frac{\tilde{S}^{(K)}}{\tilde{S}^{(K)^*}} \cdot \tilde{S}(\tilde{h})$$

(4.20)

and the minimum cost is given by:

$$f^0(\delta) = \tilde{S}^{(K)^*} \cdot \tilde{S}^{(K)} - \tilde{S}^{(K)^*} \cdot \frac{\tilde{S}^{(K)}}{\tilde{S}^{(K)^*}} \cdot \tilde{S}(\tilde{h})$$

(4.21)

(notice that $P^*(\delta) \cdot P(\delta) = \tilde{S}^{(K)^*} \cdot \tilde{S}^{(K)}$).

Then $f^0(\delta)$ is the similarity index between $s$ and $\tilde{s}$ for each $\delta$. Taking into account that $1 \leq \delta \leq \mathcal{H}$ is integer, equation (4.21) is calculated for all possible values of $\delta$, the minimum value $f^{\circ} = \text{MIN}_\delta f^0(\delta)$ is determined. The value of $\delta$ corresponding to that minimum is denotes as $\delta^\circ$. This minimum defines the best matching that can be achieved between $s$ and $\tilde{s}$. Parameters $\delta^\circ$ and $q^0$ -which is the value of $q^0$ obtained from substituting $\delta^\circ$ in (4.20)- provide, respectively, the relative displacement and rotation that allow the best matching. Note that the rotation is obtained as:

$$\phi_0 = \angle q^0$$

(4.22)
4. ACTIVE OBJECT RECOGNITION BASED ON CSS-NSSD

4.6 Using the Active Recognition Framework

The implementation of the active recognition system proposed in this chapter that, from now will be referred as CSS-NSSD system, is based on the framework developed in Chapter 3. In order to optimize the computational costs of this framework, it is introduced a modification based on the use of clusters introduced in order to reduce the computational cost during the identification process.

4.6.1 CSS-NSSD System. Implementation Details

The view database presented in Section 3.3.1 is here redefined with the following items:

**Redefining The View Database B**

1. The object label $i_l$.
2. The view label $j_l$.
3. The synthetic image of object $i_l$ taken from node $j_l$: $I_l$.
4. The feature vector $\vec{v}_l$ of the synthetic image $I_l$ corresponds to the spatial coordinates of the silhouette.
5. The NSSD descriptors of equation (4.10) are chosen as the invariant feature vector $\vec{w}_l$ of image $I_l$: $\vec{w}_l = \hat{S}(I_l)$.
6. A scalar value $X_l$ representing the level of ambiguity. This level of ambiguity (see equation (3.2) is computed by means of the clustering process. The clustering process is used in the proposed active recognition system not only to compute the level of ambiguity $X_l$, but also to reduce the computational cost of the shape recognition process.
7. A scalar that represents the accumulated evidence function $E_l$.
8. A column vector of six components which contains the accumulated Euler representation vector $\mathcal{T}_l$.

As it is well known, clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). This strategy allows us to
split the silhouette space into zones where the silhouettes are coarsely similar. Consequently, the cost in the recognition process is dramatically reduced. We have used the QT clustering method [91] in which the number of clusters is not imposed a priori. The feature vector is a NSSD composed by the second, third and penultimate descriptors. Figure 4.8 presents some examples in which it is demonstrated that taking the three main harmonics as NSSD the silhouette can be roughly reconstructed.

Let us get back to the view database $B$ definition (see Section 3.3.1). Since the feature vector $\mathbf{w}_l$ is $\hat{\mathbf{S}}^{(K^o)}_4$, from now on, we will use $\hat{\mathbf{w}}$, which is composed by the second, third and penultimate elements of $\mathbf{w}_l$ ($\hat{\mathbf{w}}=(\hat{\mathbf{S}}^{(K^o)}_4(2),\hat{\mathbf{S}}^{(K^o)}_4(3),\hat{\mathbf{S}}^{(K^o)}_4(2H-1))$).

**Figure 4.8:** Original silhouettes (red) and recovered silhouettes using the second, third and penultimate elements of NSSD descriptor vector (blue).

Suppose that, as result of the clustering process, we have a set of $R$ clusters. Let us then define $\Omega = \{c_r\} \ (1 \leq r \leq R)$ where the structure of each element $c_r$ includes:

(a) The cluster prototype $\mathbf{a}_r$,

(b) The list of the views associated to the cluster $B_r = \{b_t\}$.

**Shape Recognition**

Given the shape descriptor of the scene view $\mathbf{w}^{(k)}$, the first step in the shape recognition process consist on classifying $\hat{\mathbf{w}}^{(k)}$ in the clusters database. The classification is carry out by

$$\min_{r,\mathbf{a}} D(\mathbf{a}_r,\mathbf{w}^{(k)}) \land \mathbf{a}_r \in \Omega$$

(4.23)

which yields the number $r^o$ of the cluster which views are most similar to the scene. Thus, the set of candidates $B^{(k)}$ at step $k$, that in the active framework was computed by equation 3.3, corresponds, in the CSS-NSSD system with:
4. ACTIVE OBJECT RECOGNITION BASED ON CSS-NSSD

\[ B^{(k)} = \{ b_l \} \]

Active Strategy

Based on the results achieved by the active recognition framework, we use the heuristic model. Thus, the CSS-NSSD active recognition system is defined by the following equations:

\[
\begin{align*}
H E_i^{(k)} &= [(1 - D(\vec{v}_i, \vec{v}(k))) \cdot X_i] \forall b_l \in B^{(k)} \quad (4.25) \\
E_i^{(k)} &= \mathcal{K}(E_i^{(k-1)}, E_i^{(k)}) = E_i^{(k-1)} + H E_i^{(k)} \quad (4.26) \\
H \rho_m &= H A(\{ h_{m,l} \}) = \text{min} X_{h_{m,l}} \quad (4.27)
\end{align*}
\]

Table 4.1 shows the collection of methods used to implement the CSS-NSSD active recognition

| \( \vec{v} \)                       | shape silhouette | \( \vec{w} \)                     | NSSD descriptor | \( B^k \) | \( \{ b_l \} \), \( r^* = \text{min}_r \text{arg} D(\vec{v}_i, \vec{w}(k)) \land \vec{v}_i \in \Omega \) | \( D(w_l, w(k)) \) | \( \sqrt{\Sigma(w_l, w(k))^2} \) | \( \mathcal{D}(\{ w_l \}) \) | \( \text{min} D(\vec{w}_{h_{m,t}}, \vec{w}_{h_{m,p}}) \), 1 \leq t, p \leq q, t \neq p, q \) | \( E_i^{(k)} \) | \( H E_i^{(k)} \) | \( H \rho_m \) | \( H A \) | \( P(w_l, w(k)) \) | see section 4.5 | \( g_m = \mathcal{G}(\Theta^{(k)}, \Theta_m) \) | \( \sqrt{\Sigma(\Theta^{(k)}, \Theta_m)^2} \) where \( \Theta \) are the cartesian coordinates of \( \Theta \) robot join variables | \( \varsigma_m \) | Configured off-line considering whether the node \( m \) can be accessed by the robot end-effector | \( \mathcal{J} \) | \( \text{max}_m (\rho_m \cdot g_m \cdot \varsigma_m) \) |

**Table 4.1:** Collection of methods used to implement the proposed framework to develop a comparative analysis
4.7 Experimentation

In order to prove the effectiveness of the proposed dataset representation model, we have developed a set of experimental tests. The tests have been divided into two experiments:

1. Experiment 1: This is focused on comparing the performance of the NSSD descriptor with other popular shape descriptors.

2. Experiment 2: This is devoted to evaluating the performance of the CSS-NSSD dataset representation model in an active recognition system. A comparative analysis is developed by considering other active recognition systems.

The experimental setup consisted of a Staübli RX90 robot with a camera on the end-effector of the robot. This vision-robot system was able to capture images around the object placed in the scene. Figure 4.9(a) illustrates a typical scene with an isolated object placed on a table inside the robot workspace. The tests were carried out on a 3D Synthetic Model library (3DSL)[78]. The experiments were developed by selecting (3DSL) 18 free form objects from this library. The set of object models can be observed in Figure 4.9(b), whereas Figure 4.9(c) shows samples of images captured in the robot-vision system and later used during the tests. Note that the background is uniform and the objects’ illumination changes according to the sensor position.

![Figure 4.9](image-url): Experimental setup. a) The experimental platform uses a Staübli robot with a camera on the end-effector. b) Synthetic objects in the dataset. c) Samples of images captured in the vision-robot setup and used during the active recognition tests.
4. ACTIVE OBJECT RECOGNITION BASED ON CSS-NSSD

4.7.1 Experiment 1

Since shape recognition is a key process for any 3D recognition system based on object appearance, the first tests were focused on measuring the performance of our suggested shape representation model (NSSD). A comparative analysis was carried out by taking the following parameters into consideration: recognition rates ($R$), computational cost ($C$) and shape pose estimation error ($P$). The pose estimation error was measured as a mean square error between the query shape and the one identified in the dataset after being transformed according to the estimated pose parameter. In this test, we compared the behavior of the NSSD descriptor with other popular descriptors based on shape contours (Fourier descriptor (FD) [39], Principal Component Analysis (PCA) [96] and Boundary Moment (BM)[43]), and based on shape region (Zernike Moments (ZM)[44], Complex Moments (CM)[46]).

For each test, the number of elements of the shape descriptor was: 64 (NSSD), 64 (FD), 7 (BM), 8 (PCA), 12 (ZM) and 11 (CM). These numbers were established after testing the methods for a different number of elements, and, eventually, choosing those which yielded the best shape recognition rates.

The shape recognition process was developed by using the Euclidean distance as a similarity measure. The shape recognition test was accomplished by capturing one image from a camera located at the end-effector of a robot. We took a total of 86 images for this test. The recognition rate is shown in Figure 4.10(a), the pose error is measured in pixels in Figure 4.10(b) and the computational cost is evaluated in seconds in Figure 4.10(c). In general, NSSD descriptors yield the best tradeoff between the three evaluation parameters. The recognition rate is acceptable and comparable with the highest rates provided by PCA and CM. It is by far the most accurate technique for pose estimation, and has an acceptable computational cost. Some specific comments follows.

- Although the computational cost of NSSD is higher than that of the classical Fourier Descriptor, the recognition rate and pose estimation show a better performance.

- The improvement of NSSD on the traditional Fourier Descriptor is owing to the fact that NSSD is more robust to shape variations (viewpoint variation, noise, segmentation error). Only the Complex Moments descriptor yields higher recognition rates than NSSD. However, it is important to bear in mind that if the object recognition application requires high accuracy of the pose, then the NSSD descriptor is better than the Complex Moment descriptor.
4.7 Experimentation

Figure 4.10: NSSD descriptor performance. (a) Recognition rates. (b) Shape pose error. (c) Computational cost.

4.7.2 Experiment 2

The tests developed in this section are focused on quantifying the performance of the proposed dataset in an active recognition system. We evaluate several parameters: recognition rate \((R)\), computational efficiency \((C)\) and pose estimation error \((P)\). Note that, in this case, the parameters refer to the 3D object. The first two parameters can be evaluated by using the robotic platform but the pose estimation error must be evaluated by means of a simulator.
The selection of a set of active recognition systems (ARS) have been done considering different aspects:

- **Object models (OM):** aspect graph, salient views, CSS

- **Shape representation schemes (SR):** parametric eigenspaces (PCA), NSSD descriptor, Fourier Descriptors

- **Active strategies (AS):** taking the next view to minimize an ambiguity function and incorporating explicit planning algorithms.

- **Methods for representing uncertainty (MU):** probabilistic, Dempster-Shafer theory, Heuristic.

We have chosen seven active recognition systems, which will be denoted as AR (active recognition). AR1 corresponds with the active recognition system developed in the Experiments 2 in Chapter 3 (see Table 3.3). AR2 is a variant of the active recognition system presented in this chapter in which the objects are modeled by salient views [109]; AR3 is the proposed CSS-NSSD system; AR4 is the technique published by Borosting et al. in [82]. In Borotschnig et al. [82] three different strategies are compared using stochastic models. The results from this comparative study show that the probabilistic model is the best choice. We have also implemented this probabilistic model. In our case, a set of training views and the representation of the information of this views by means of PCA [96] is necessary. AR5 is the system developed by Hutchinson and Kak [88]. It is stochastic too but the objects are modeled by an aspect graph. AR6 corresponds to an implementation of the framework proposed by Kovacik et al. [89] in which the active strategy is based on explicit planning. Finally, AR7 [100] is a deterministic active recognition system.

Table 4.2 summarizes the set of active recognition methods used in this experiment. The first notable result is that the CSS model reduced the number of views of the original dataset to 63%, whereas [109] reduced them to 78%.

### 4.7.2.1 Recognition and computational evaluation on a robotic platform

Table 4.3 shows the comparison between different shape recognition systems with regard to recognition rate ($R$) and computational efficiency ($C$). In this case, the computational cost ($C$)
4.7 Experimentation

<table>
<thead>
<tr>
<th>ID</th>
<th>ARS</th>
<th>OM</th>
<th>SR</th>
<th>AS</th>
<th>MU</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR1</td>
<td>Table 3.3</td>
<td>all sphere views</td>
<td>FD</td>
<td>ambiguity minimization</td>
<td>Heuristic</td>
</tr>
<tr>
<td>AR2</td>
<td>CSS-NSSD</td>
<td>salient views [109]</td>
<td>NSSD</td>
<td>ambiguity minimization</td>
<td>Heuristic</td>
</tr>
<tr>
<td>AR3</td>
<td>CSS-NSSD</td>
<td>CSS</td>
<td>NSSD</td>
<td>ambiguity minimization</td>
<td>heuristic</td>
</tr>
<tr>
<td>AR4</td>
<td>[82]</td>
<td>all sphere views</td>
<td>PCA</td>
<td>ambiguity minimization</td>
<td>Probabilistic</td>
</tr>
<tr>
<td>AR5</td>
<td>[88]</td>
<td>aspect graph</td>
<td>PCA</td>
<td>ambiguity minimization</td>
<td>Dempster Shafer</td>
</tr>
<tr>
<td>AR6</td>
<td>[89]</td>
<td>CSS</td>
<td>NSSD</td>
<td>explicit planning</td>
<td>-</td>
</tr>
<tr>
<td>AR7</td>
<td>[100]</td>
<td>all sphere views</td>
<td>FD</td>
<td>ambiguity minimization</td>
<td>Heuristic</td>
</tr>
</tbody>
</table>

Table 4.2: Set of active recognition systems (ARS) and the main methods used by their main modules

<table>
<thead>
<tr>
<th>Active Recognition System</th>
<th>( T (s) )</th>
<th>( W )</th>
<th>( C(s) )</th>
<th>( R (%) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AR1)</td>
<td>0.20</td>
<td>5.5</td>
<td>1.10</td>
<td>94</td>
</tr>
<tr>
<td>(AR2)</td>
<td>0.17</td>
<td>5.2</td>
<td>0.88</td>
<td>91</td>
</tr>
<tr>
<td>(AR3)</td>
<td>0.18</td>
<td>3.8</td>
<td>0.68</td>
<td>95</td>
</tr>
<tr>
<td>(AR4)</td>
<td>0.21</td>
<td>4.6</td>
<td>0.86</td>
<td>96</td>
</tr>
<tr>
<td>(AR5)</td>
<td>0.19</td>
<td>4.3</td>
<td>0.82</td>
<td>93</td>
</tr>
<tr>
<td>(AR6)</td>
<td>0.14</td>
<td>3.1</td>
<td>0.43</td>
<td>78</td>
</tr>
<tr>
<td>(AR7)</td>
<td>0.17</td>
<td>4.1</td>
<td>0.70</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 4.3: Comparison between different active recognition systems on the robotic platform

is computed by \( C = W \cdot T \), where \( W \) is the mean number of sensor positions and \( T \) is the computational cost at each sensor position.

Comparing the CSS-NSSD system (AR3) with the set of active recognition systems from Table 4.3 we can conclude:

- The AR3 system improves the results yielded by the AR1 system. Note that the number of sensor positions is reduced in AR3 and the computational efficiency of this system is higher since it requires a smaller number of sensor positions. This result also proves that our representation model is more robust than the original method AR1, in which the shape recognition system identifies a smaller number of candidates in each iteration.

- AR2 uses 15% less views than AR3, and the object pose estimation is worse. This behavior is related to the uncertainty that is present during the hypothesis estimation.
as a result of the differences between the scene view and the salient view. More robot movements are therefore required to solve the uncertainty problem.

- AR4 and AR5 are samples of active recognition systems based on probabilistic methodologies. The performances of AR4, AR5 and AR3 are very similar. Bear in mind that in the case of AR3 it is not necessary to use a training step.

- AR6 system uses the same dataset representation that AR3 but the recognition rates demonstrate the low performance of an explicit planning method to develop the active strategy.

- AR3 system outperforms AR7 because it is more robust to ambiguity and uncertainty problems.

### 4.7.2.2 Pose error evaluation in a simulator

The experiments developed in the robotic setup cannot evaluate the object pose in a precise manner. Nevertheless, this parameter can be evaluated by using a simulator. In order to evaluate the pose for AR1, AR2, AR3, AR4, AR5, AR6 and AR7, we have carried out a set of tests in a simulated framework.

The simulator takes views from 3D synthetic models after a random transformation. The original model \( o \) is thus transformed into the rotated model \( \hat{o} \). The simulator also provides the projection of \( \hat{o} \) from the viewpoint corresponding to the current sensor position. The synthetic image from the projection is then used by the active recognition systems to recognize the object and to estimate the pose. The pose accuracy parameter \( (P) \) is computed by means of the mean square error between the rotated object \( (\hat{o}) \) and the object after being transformed by means of the estimated pose parameters \( (\hat{\theta}) \).

Let \( \hat{x}_i, \hat{y}_i, \) and \( \hat{z}_i \) be the coordinates associated with the cloud of points for the sample \( (\hat{o}_i) \) and \( \hat{x}, \hat{y}, \) and \( \hat{z} \) be the coordinates of the candidate after applying a transformation \( \hat{T} \) estimated by the active recognition system. The pose error is computed by:

\[
P_e = \sqrt{\frac{\sum (\hat{x}_i - \hat{x})^2 + (\hat{y}_i - \hat{y})^2 + (\hat{z}_i - \hat{z})^2}{\hat{n}}} \tag{4.28}
\]

where \( \hat{n} \) is the number of points in the sample. See [100] for more information about the rotation process.
Table 4.4 shows the computational cost ($C$), recognition rate ($R$), and pose accuracy ($P_e$) results, obtained in the simulator for 54 objects which were rotated randomly.

<table>
<thead>
<tr>
<th>Active model</th>
<th>$C$(s)</th>
<th>$R$ (%)</th>
<th>$P_e$(cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AR1)</td>
<td>0.78</td>
<td>95</td>
<td>0.0158</td>
</tr>
<tr>
<td>(AR2)</td>
<td>0.67</td>
<td>94</td>
<td>0.0306</td>
</tr>
<tr>
<td>(AR3)</td>
<td>0.70</td>
<td>99</td>
<td>0.00121</td>
</tr>
<tr>
<td>(AR4)</td>
<td>0.73</td>
<td>99</td>
<td>0.0244</td>
</tr>
<tr>
<td>(AR5)</td>
<td>0.70</td>
<td>98</td>
<td>0.0285</td>
</tr>
<tr>
<td>(AR6)</td>
<td>0.31</td>
<td>93</td>
<td>0.00302</td>
</tr>
<tr>
<td>(AR7)</td>
<td>0.69</td>
<td>96</td>
<td>0.1004</td>
</tr>
</tbody>
</table>

Table 4.4: Comparison between different active recognition systems in simulation

Table 4.4 shows that AR3 has better performance than the other active recognition systems. Note that AR3 gives the best results both in recognition rate and in pose estimation accuracy.

### 4.8 Final Discussions and Conclusions

This chapter has developed an implementation of an active recognition system based on a dataset representation model. The objective is to normalize and reduce the object representation model with the aim of making the chosen active recognition algorithm as efficient, accurate and fast as possible. To achieve this goal we propose a representation model based on two concepts: Canonical Sphere and Normalized Silhouette in the Fourier Spectral Domain.

The Canonical Sphere Section (CSS) makes it possible to highly reduce a view-based object database when objects with reflective symmetry properties are present in the database. As was explained in Section 2, the initial number of viewpoints (from which an object can be totally described) can be drastically reduced to a few viewpoints. CSS therefore contains the viewpoints which are necessary to characterize an object, thus eliminating the redundant information, and consequently simplifying the model representation and the database.

The proposed NSSD descriptor is, meanwhile, capable of normalizing the shape (silhouette) of the projected image of the object in the spectral domain, so that the normalized shape is invariant to affine transformations. This makes NSSD robust to small 2D shape deformations and geometric transformations. NSSD also represents, with the same accuracy, any shape by using the same number of elements in the descriptor.
The good performance of the CSS-NSSD model has been experimentally proved. We have carried out two tests. In Test 1, a comparison of the performance of the NSSD descriptor with other popular shape descriptors took place. Test 2 evaluates the performance of the CSS-NSSD representation model in active recognition systems. In this case, a comparative analysis considering four active recognition systems was carried out. In conclusion, we have proven that CSS-NSSD models lead to efficient, accurate, computationally low cost active recognition systems.
A Grasping Experience with the CSS-NSSD Approach

The goal of this chapter is to study the performance of the proposed active recognition model when it is integrated as a part of an intelligent robotic task. The recognition system particularly helps the robot to carry out a reliable grasping activity.

5.1 Introduction

Most manipulation systems currently perform accurate manipulation tasks by using computer vision as a feedback tool. Vision systems used as control tools (frequently called visual servoing systems) have become very popular in vision-robot developments over the last two decades [110, 111]. They provide visual perception of the scene and really help robots to accomplish accurate object manipulation tasks. The simplest vision system consists of a single camera coupled in the end-effector of the robot together with the corresponding computer vision algorithms. More complicated cases, which include stereo and multi-synce vision, range finders, and laser scanners can also be found in literature.

In general it can be said that visual control techniques for manipulation tasks promise substantial advantages when working with targets whose positions are unknown. They can also be extended to many other applications in fields such as manufacturing, service robots, fruit-picking robots, robotic ping-pong, juggling, balancing, etc. A good review of the different approaches can be found in surveys [112, 113, 114].
Two main subjects must be taken into account in this framework: the vision-robot configuration and the visual control model.

There are mainly two types of vision-robot configurations: the eye-in-hand configuration and the stand-alone configuration [112, 115]. In the first case, the camera(s) are mounted on the end-effector of the robot and move(s) as the robot does. This configuration makes it possible to obtain a more detailed view of the object of interest. In the second case, the camera is fixed on the workspace of the robot. This configuration provides a wider field of vision of the scene.

The use of a single camera in an eye-in-hand configuration has been a very common setup in many reported works [115]. In this case, the hand-eye calibration—that is, the transformation between the end-effector and the camera coordinate frames—is assumed to be known. Several works with this configuration using image features [116, 117] or model-based (117, 118) tracking techniques can be found in literature. Stand-alone configurations with a single camera were usual in the early systems [118]. However, some recent works which use this kind of approach also exist [119, 120].

In stereo systems, the usual approach consists of estimating the disparity, which yields the depth of the scene [121, 122]. However, one of the principal problems with this is the detection of matching features between two or more images. The use of a stereo head-mounted camera at the robot’s end-effector is less common than in a stand-alone configuration. Note that, in the latter case, the baseline (i.e., the distance between the cameras) could be made sufficiently long to obtain accurate depth estimation [115]. Some examples of two cameras in a stand-alone configuration can be found in [123, 124]. Finally, we can find systems which use the stereo head in an eye-in-hand configuration, like those described in [122, 125].

5.2 Visual Control

The primary goal of the visual control process in a grasping application is to guide the gripper towards the object in the scene. This process could be carried out in a closed-loop or an open-loop manner.

5.2.1 Open loop Robot Control or Look-And-Move

The extraction of image information and the control of a robot are two separate tasks if image processing is performed first, followed by the generation of a control sequence. A typical
example is to recognize the object to be manipulated by matching image features to a model of the object and computing its position and orientation (pose) relative to the camera (or to the robot) coordinate system. This pose (Cartesian-space information) is used to move the robot to the desired pose relative to the object. If the pose of the object is to be estimated, then the model of the object must be available. In order to move the robot based on the visual information extracted in the camera frame, the camera(s) must be calibrated with regard to the robot. In addition, the robot direct and inverse kinematic models must be available to convert Cartesian-space robot positions into joint-space configurations. The robot can then execute the task by performing 'blind' movements, which assumes that the environment remains static after the robot has started to move (open-loop approach).

The main reason why open loop controls are used is the relatively low sampling rates available from vision, which make direct control of the robot with complex nonlinear dynamics an extremely challenging control problem. Secondly, many robots already have an interface that can accept Cartesian velocity or incremental position commands. This simplifies the construction of the visual servo system and also makes the methods more portable. Thirdly, look and move separates the kinematic singularities of the mechanism from the visual controller, thus allowing the robot to be considered as an ideal Cartesian motion device. One drawback of the open loop approach is the need for an accurate calibration of the camera-manipulator system, and the fact that the accuracy of the resulting operation depends directly on the accuracy of the visual sensor and manipulator. An example of such a control structure, when used for grasping an object, is provided by Kragic et al.[120].

The benefits and drawbacks of open-loop robot control approaches are listed below.

**Benefits:**

- No real time vision system is required, hence the timing of the arm control is not dependent on the vision system.
- The arm control and the vision system can be encapsulated, thus making them more portable.
- An algebraic inverse kinematic transform already exists for the manipulator used.
- The handling of kinematic singularities will be carried out by the algebraic inverse kinematic algorithm.
5. A GRASPING EXPERIENCE WITH THE CSS-NSSD APPROACH

- The vision system can be used for other purposes such as self-localization or state recognition (door open or closed).

**Drawbacks:**

- An accurate calibration of the camera-manipulator system is required.
- A vision system with high accuracy is required.
- An object model is needed.

5.2.2 Closed Loop Robot Control (Visual Servoing)

Conversely, systems that observe both the end-effector and target features can perform with an accuracy that is independent of hand-eye calibration error. Note also that closed loop systems can easily deal with tasks that involve the positioning of objects in the end-effector, whereas open loop systems must use an inferred object location. From a theoretical perspective it would appear that closed loop systems would always be preferable to open loop systems.

However, since closed loop systems must track both the end-effector and the target object, the implementation of a closed loop controller often requires the solution of a more demanding vision problem. These endpoint closed-loop vision control systems are called “visual servoing”. This term was introduced by Hill and Park in [15] to distinguish their approach from earlier experiments in which the system alternated between picture taking and moving. Visual servoing can be further distinguished between:

- **Position-based visual servo systems (PBVS).** The error is computed in 3D Cartesian Space, and both the current and the desired pose of the robot must be expressed in this space. These systems use the image features to perform an estimation of the current pose of the object of interest, usually with regard to the camera-attached coordinate system. The computation of this estimation often requires knowledge of the internal parameters of the camera and, in some cases, a model of the object. In this type of systems, the task function is also referred to as kinematic error function [112] or virtual kinematic constraint [112, 126].

- **Image-based visual servo systems (IBVS).** Sanderson and Weiss actually proposed the term ‘visual servo’ for this type of systems.
5.3 Grasping Planner

However, Hutchinson et al. [112] suggested the term Image-based visual servo systems (IBVS), since ‘visual servo’ has been widely used to refer to any closed-loop vision-based control system. In IBVS systems, the error is directly computed in image space. Both the target pose of the robot and the current pose are expressed in image space. In these systems, the task function is also called the image error function [112].

The benefits and drawbacks of the closed-loop robot control approach are the following:

**Benefits:**

- The visual data can compensate for manipulator positioning inaccuracies and sensor noise.
- Less accuracy is needed in the vision system.

**Drawbacks:**

- A real-time vision system is required, because the arm control directly depends on the vision system.
- An eye-in-hand camera robot configuration is required.
- The controller has to take care of the arm singularities by itself.
- The control of manipulator and the vision system are tightly coupled

5.3 Grasping Planner

5.3.1 An Overview

A hand-eye configuration with open-loop control strategy has been tested in this thesis. One camera is adapted at the tip of the manipulator, which is also equipped with a two finger gripper with the objective of picking up and placing one isolated object. In the test, it is assumed that the object remains still in the scene until the grasping takes place, otherwise an open-loop control does not make any sense.

Figure 5.1 shows an overview of all the agents involved in the grasping procedure. The recognition and positioning of the object using the proposed active system play the main role in the experimentation throughout all the stages involved. The objective is to integrate the
5. A GRASPING EXPERIENCE WITH THE CSS-NSSD APPROACH

recognition system into the robot, and prove its efficiency and accuracy for grasping tasks. In general, the setup can be summarized in the four blocks shown in 5.1.

The first block concerns the sensory processing. In this stage, the camera is pointed towards the object placed in the scene, it takes the image and performs the processes needed to carry out another sample, if necessary. As was explained in Chapter 3, the active recognition system provides the next best view of the camera until the recognition is complete. This occurs when a view hypothesis $\hat{I}_o$ reaches a given level $\eta$ of accumulated evidence.

In the second block, the grasping synthesis is carried out. The contact points of the recognized object are established by means of the algorithm of Adán et.al [127]. This is a 3D object-model based grasping solution which calculates a sorted list with the coordinates of the best pairs of contact points. Taking into account the pose of the object with regard to the robot base, these coordinates are then translated to the robot base reference system by using the result of equation 3.24.

Assuming a candidate pair of contact points, the inverse kinematic and path planning stages calculate the wrist pose and the movement of the arm of the robot from the home position to the grasping position. Although the scene is free of collisions with other objects, the shape and

![Grasping planner outline.](image)

**Figure 5.1:** Grasping planner outline.
5.3 Grasping Planner

size of the gripper itself may influence the best grasping solution owing to collision with the surface on which the object is located. The inverse kinematics at the grasping location might also be unsuccessful. In all these cases, the candidate pair of points is rejected and a new pair from the list is then checked. The path planning algorithm guarantees a smooth and reliable path in the scenario.

When the two last stages converge, the grasping is executed. The robot control then carries out the movement of the robot links, providing the necessary power to follow the path calculated. After that, the gripper opens and closes its fingers and eventually carries the object to a repository. It is clear that the grasping action consists of multiple steps, the object recognition part being only one of them. The other processes involved (i.e. inverse kinematics, path planning, wrist configurations, control of the manipulator) can be carried out, under the supervision of the user, by conventional programming and software on board industrial robots. Since these steps are not within the scope of this thesis, only the minimum necessary information about them is included. Some of those phases are briefly described in the following sub-sections, and grasping synthesis, kinematics and path planning processes in particular are discussed.

5.3.2 Grasping synthesis: contact model and grasp-points coordinates.

Grasping synthesis is highly influenced by assumptions and constraints. For instance, mechanical design aspects, such as the number of fingers and the degrees of freedom of each finger, clearly influence the approach to the problem. Anthropomorphic hands ([128, 129]) represent the most complex approach in terms of grasping control and planning. However, many works [130] have focused on reduced-complexity for specific and multipurpose hands. In fact, minimalistic designed grippers like ([131, 132]) have proved to be useful in multipurpose tasks.

Following a minimalistic point of view, in this work we use a 2-fingered parallel gripper. As is demonstrated in the experimental results of Section 5.4, our grasping planner allows this gripper to grasp a variety of objects, making it suitable for multipurpose manipulation environments.

Following a minimalistic point of view, in this work we use a 2-fingered parallel gripper. As is demonstrated in the experimental results of Section 5.4, our grasping planner allows this gripper to grasp a variety of objects, making it suitable for multipurpose manipulation environments.
5. A GRASPING EXPERIENCE WITH THE CSS-NSSD APPROACH

Physical model assumptions, such as the contact model or the prehension model, also influence the grasping solution. A standard classification of contact models distinguishes between [133]:

- **Frictionless point contact.** The finger can only exert a normal force through the point of contact.
- **Hard-finger contact.** This is a point contact with friction. The finger can exert any force pointing into the friction cone and the point of contact.
- **Soft-finger contact.** The friction over the area of contact allows the finger to exert pure torques in addition to pure forces. The finger can exert torques in both directions, about the normal axis at the point of contact.

This thesis takes the soft-finger contact model. By definition, each soft-finger contact can be approximated by a four-wrench convex and the force-closure can be obtained with at least two soft-finger contacts if and only if the angle $\beta$ between the two plans of contact is strictly less than the angle of the friction cone $2\delta$ (Fig 5.2(a)). The same force-closure condition can be found in [127].

The contact points have been obtained by following the strategy of Adán et. al, published in [127]. This is a 3D object-model based grasping solution. The grasping solution essentially lies in the use of a simplified object representation model to define a set of non-strict local properties of the contact-points. These features allow us to evaluate important aspects around the hypothetic contact-points (e.g. extended curvature, relative location on the surface, neighbor normal vectors, safety distance, etc), along with the global properties of the object (e.g. principal directions, dimension, etc), and to then generate a set of grasp starting positions. On the whole, the huge number of possible hand configurations is reduced to a few reliable grasp candidates.

For our soft-finger contact model, the line crossing a candidate pair of contact points $P_1$ and $P_2$ must be inside the friction cones, signifying that $\beta/2 < \delta$, where $\tan\delta = \mu$. This ensures the force-closure condition. The Adan method is used to calculate a set of optimal grasp points for two-finger grippers, which are ordered according to a quality function, whilst maintaining the requirement safety of the grasp. Figure 5.2(b) shows examples of optimal grasps.

To obtain the final position and orientation of the gripper, the coordinates of a pair of contact points $(P_1, P_2)$ (in the object model frame $(^bP_1, ^bP_2)$) are transformed into coordinates in the
5.3 Grasping Planner

![Figure 5.2: a) Soft-finger contact b) Pairs of optimal contact points obtained by following the Adan's algorithm.](image)

Figure 5.2: a) Soft-finger contact b) Pairs of optimal contact points obtained by following the Adan’s algorithm.

robot base frame \((R_b P_1, R_b P_2)\) by means of transformation \(T_{L_0}\) in equation (3.24). Remember that \(T_{L_0}\) was the homogeneous transform between the frame in the center of the tessellated frame \((S_b)\) that surrounded the object and the robot base frame \((R_b)\). Therefore, \(T_{L_0}\) represents the estimated pose of the object associated with view \(\hat{L}_o\) with regard to the robot base. The coordinates of the contact points in the robot reference system are thus obtained in equation (5.1).

\[
R_b P_i = T_{L_0}(S_b P_i), \ i = 1, 2
\]  

(5.1)

From here on, we will take the nomenclature \(P_i\) rather than \(R_b P_i\), always assuming coordinates in the base robot frame.

5.3.3 Kinematics of the robot

Direct kinematics

The manipulation test was performed on a PUMA industrial robot. This is an articulated arm with 6 degrees of freedom in which the robot controllers provides the absolute position of the robot at all times. In our case, the closing and opening of the grasp supposes one additional degree of freedom. A commonly used convention for selecting frames of reference in robotic applications is the Denavit-Hartenberg, or D-H convention. In this convention, each
5. A GRASPING EXPERIENCE WITH THE CSS-NSSD APPROACH

A homogeneous transformation is represented as a product of basic transformations. The robot and the D-H kinematic parameters of the robot are shown in Figure 5.3 and Table 5.1, in which it is assumed that the sixth link corresponds to the two finger gripper.

![Figure 5.3: Frames of PUMA robot.](image)

<table>
<thead>
<tr>
<th>Link</th>
<th>( a_i - 1 )</th>
<th>( a_{i-1} )</th>
<th>( d )</th>
<th>( \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( \theta_1 )</td>
</tr>
<tr>
<td>2</td>
<td>-90</td>
<td>0</td>
<td>0</td>
<td>( \theta_2 )</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>( a_2=450\text{mm} )</td>
<td>0</td>
<td>( \theta_3 )</td>
</tr>
<tr>
<td>4</td>
<td>-90</td>
<td>0</td>
<td>( 450\text{mm} )</td>
<td>( \theta_4 )</td>
</tr>
<tr>
<td>5</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>( \theta_5 )</td>
</tr>
<tr>
<td>6</td>
<td>-90</td>
<td>0</td>
<td>( d_6 = 250\text{mm} )</td>
<td>( \theta_6 )</td>
</tr>
</tbody>
</table>

Table 5.1: DH parameters

If the D-H parameters are known, the direct kinematics provides the total homogeneous transform \( R_{b}^{R_{t}} T \), \( (R_t = \text{robot tip frame and } R_b = \text{robot base frame}) \)

\[
R_{b}^{R_{t}} T = R_{b}^{R_{t}} T_{1}^{R_{t}} T_{2}^{R_{t}} T_{3}^{R_{t}} T_{4}^{R_{t}} T_{5}^{R_{t}} T_{6}^{R_{t}} T
\]

(5.2)
5.3 Grasping Planner

\[
^{Rb}_{1}T = \begin{bmatrix}
\cos \theta_1 & -\sin \theta_1 & 0 & 0 \\
\sin \theta_1 & \cos \theta_1 & 0 & 0 \\
0 & 0 & 1 & d_3 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (5.3)

\[
^{1}2T = \begin{bmatrix}
\cos \theta_2 & -\sin \theta_2 & 0 & 0 \\
0 & 0 & 1 & d_3 \\
-\sin \theta_2 & -\cos \theta_2 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (5.4)

\[
^{2}_{3}T = \begin{bmatrix}
\cos \theta_3 & -\sin \theta_3 & a_2 & 0 \\
\sin \theta_3 & \cos \theta_3 & 0 & 0 \\
0 & 0 & 1 & d_3 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (5.5)

\[
^{3}_{4}T = \begin{bmatrix}
\cos \theta_4 & -\sin \theta_4 & a_2 & 0 \\
0 & 0 & 1 & d_4 \\
-\sin \theta_4 & -\cos \theta_4 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (5.6)

\[
^{4}_{5}T = \begin{bmatrix}
\cos \theta_5 & -\sin \theta_5 & 0 & 0 \\
0 & 0 & -1 & 0 \\
\sin \theta_5 & \cos \theta_5 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (5.7)

\[
^{5}_{6}T = \begin{bmatrix}
\cos \theta_6 & -\sin \theta_6 & 0 & 0 \\
0 & 0 & 1 & d_6 \\
-\sin \theta_6 & -\cos \theta_6 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (5.8)

Taking \(d_3 = 0\) and \(a_3 = 0\), the components of matrix \(^{Rb}_{Rt}T\) are the followings:

\[
^{Rb}_{Rt}T = \begin{bmatrix}
n_x & s_x & a_x & p_x \\
n_y & s_y & a_y & p_y \\
n_z & s_z & a_z & p_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (5.9)

\[
n_x = \cos \theta_1[(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3)(\cos \theta_4 \cos \theta_5 \cos \theta_6 - \sin \theta_4 \sin \theta_6) - (\cos \theta_2 \sin \theta_3 + \\
+ \sin \theta_2 \cos \theta_3) \sin \theta_5 \cos \theta_6] + \sin \theta_1(\sin \theta - 4 \cos \theta_4 \cos \theta_6 + \cos \theta_4 \sin \theta_6)
\] (5.10)

\[
s_x = \cos \theta_1[(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3)(-\cos \theta_4 \cos \theta_5 \cos \theta_6 - \sin \theta_4 \sin \theta_6) - (\cos \theta_2 \sin \theta_3 + \\
+ \sin \theta_2 \cos \theta_3) \sin \theta_5 \cos \theta_6] + \sin \theta_1(\sin \theta - 4 \cos \theta_4 \cos \theta_6 - \sin \theta_4 \cos \theta_4 \sin \theta_6)
\] (5.11)
5. A GRASPING EXPERIENCE WITH THE CSS-NSSD APPROACH

\[
a_z = - \cos \theta_1 \left[ \left( \cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3 \right) \cos \theta_4 \cos \theta_5 + \left( \cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3 \right) \cos \theta_5 \right] - \\
- \sin \theta_1 \sin \theta_4 \sin \theta_5 \tag{5.12}
\]

\[
p_x = \cos \theta_1 \left[ \left( a_2 \cos \theta_2 - d_4 \left( \cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3 \right) \right) \right] \tag{5.13}
\]

\[
n_y = \sin \theta_1 \left[ \left( \cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3 \right) \cos \theta_4 \cos \theta_5 \cos \theta_6 - \sin \theta_4 \sin \theta_6 \right] - \\
\left( \cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3 \right) \sin \theta_5 \cos \theta_6 - \cos \theta_1 \left( \cos \theta_4 \cos \theta_6 - \sin \theta_4 \sin \theta_6 \right) \tag{5.14}
\]

\[
s_y = \sin \theta_1 \left[ \left( \cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3 \right) \left( - \cos \theta_4 \cos \theta_5 \cos \theta_6 - \sin \theta_4 \sin \theta_6 \right) - \left( \cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3 \right) \sin \theta_5 \cos \theta_6 \right] - \\
\cos \theta_1 \left( \cos \theta_4 \cos \theta_6 - \sin \theta_4 \sin \theta_6 \right) \tag{5.15}
\]

\[
a_y = - \sin \theta_1 \left[ \left( \cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3 \right) \cos \theta_4 \cos \theta_5 + \left( \cos \theta_2 sin \theta_3 + \right. \right. \\
+ \left( \sin \theta_2 \cos \theta_3 \right) \cos \theta_5 \right] + \sin \theta_1 \sin \theta_4 \sin \theta_5 \tag{5.16}
\]

\[
p_y = \sin \theta_1 \left[ \left( a_2 \cos \theta_2 - d_4 \left( \cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3 \right) \right) \right] \tag{5.17}
\]

\[
n_z = - \left( \cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3 \right) \left( \cos \theta_4 \cos \theta_5 \cos \theta_6 - \sin \theta_4 \sin \theta_6 \right) - \\
- \left( \cos \theta_2 \cos \theta_3 - \sin \theta_2 \cos \theta_3 \right) \sin \theta_5 \cos \theta_6 \tag{5.18}
\]

\[
s_z = - \left( \cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3 \right) \left( - \cos \theta_4 \cos \theta_3 \cos \theta_6 - \sin \theta_4 \sin \theta_6 \right) + \\
+ \left( \cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3 \right) \sin \theta_5 \sin \theta_6 \tag{5.19}
\]

\[
a_z = \left( \cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3 \right) \cos \theta_4 \sin \theta_5 - \left( \cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3 \right) \cos \theta_5 \tag{5.20}
\]
5.3 Grasping Planner

\[ p_z = a_2 \sin \theta_2 - d_4 (\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) \]  \hspace{1cm} (5.21)

**Inverse kinematics for grasping**

The robot tip frame at the grasping position \( R_t \) is defined from the contact point coordinates (in the base robot reference system). Let us assume the two-finger gripper shown in Figure 5.4.

**Figure 5.4:** Defining the robot tip frame at grasping position.

Origin \( O_t \) coincides with the contact-points’s centroid, axis \( Y_6 \) is defined by the unitary vector \( \vec{s} \) in the direction followed by \( \vec{P_1P_2} \), and axes \( X_6 \) and \( Z_6 \) lie in the perpendicular plane \( \Pi_\vec{s} \). In order to avoid potential collisions of the gripper with the surface on which the object is located, axis \( Z_6 \) is defined by the projection of unitary vector \( \vec{u_z} \) on the plane \( \Pi_\vec{s} \). The unitary vectors \( \vec{n}, \vec{s}, \vec{a} \) corresponding to axes \( X_6, Y_6, Z_6 \) and position \( \vec{p} \) of \( O_t \) in the base robot frame, are formally the following:

\[ \vec{s} = \frac{\vec{P_1P_2}}{||\vec{P_1P_2}||} \]  \hspace{1cm} (5.22)

\[ \vec{a} = \varphi(\vec{u_z}, \Pi_\vec{s}) \]  \hspace{1cm} (5.23)

\[ \vec{n} = \vec{s} \times \vec{a} \]  \hspace{1cm} (5.24)

\[ \vec{p} = (\vec{P_1} + \vec{P_2})/2 \]  \hspace{1cm} (5.25)
5. A GRASPING EXPERIENCE WITH THE CSS-NSSD APPROACH

where $\varphi(\vec{u}_Z, \Pi_3)$ is the function that calculates the unitary vector of the projection of $\vec{u}_Z$ on $\Pi_3$. Therefore $R_b^r T$ is known and the inverse kinematics calculates the set of joints $\Theta = (\theta_1, \ldots, \theta_6)$.

The equation is then solved:

$$R_b^r T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = R_b^r T(\theta_1) \cdot T(\theta_2) \cdot T(\theta_3) \cdot T(\theta_4) \cdot T(\theta_5) \cdot T(\theta_6)$$

(5.26)

The robot’s joint configuration signifies that the end-effector may be placed on the opposite side so that up to eight solutions can be carried out. The opposite configuration is obtained from equations (5.27):

$$\theta_4' = \theta_4 + 180^\circ, \quad \theta_5' = \theta_5, \quad \theta_6' = \theta_6 + 180^\circ$$

(5.27)

An algebraic solution can be obtained from equations 5.10 to 5.21. The joint coordinates are calculated as follows:

$$\theta_1 = \arctan(p_y, p_x) - \arctan2(d_3, \pm \sqrt{p_x^2 + p_y^2 - d_3^2})$$

(5.28)

$$\theta_3 = \arctan(a_3, d_4) - \arctan2(K, \pm \sqrt{a_3^2 + d_4^2 - K^2})$$

(5.29)

where

$$K = \frac{p_x^2 + p_y^2 + p_z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2}$$

(5.30)

and $\arctan$ computes the arctangent.

There are two solutions for $\theta_1$ and $\theta_3$. Meanwhile, $\theta_2$ is calculated from $\theta_1$ and $\theta_3$.

$$\theta_2 = \arctan\left[(-a_3 - a_2\cos\theta_3)p_z - (\cos\theta_1 p_x + \sin\theta_1 p_y)(a_2\sin\theta_3), (a_2\sin\theta_3 - d_4)p_z - (a_3 + a_2\cos\theta_3\sin\theta_3)(\cos\theta_1 p_x + \sin\theta_1 p_y)\right] - \theta_3$$

(5.31)

Note that $\theta_2$ has four possible solutions.

$$\theta_4 = \arctan(-a_s)$$

(5.32)

$$\theta_5 = \arctan(-a_s)$$

(5.33)
5.3 Grasping Planner

\[ \theta_6 = \arctan(h, m) \]  

(5.34)

where

\[ h = \left(-r_{11} \left(\cos \theta_1 \left(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3\right) \sin \theta_4 - \sin \theta_1 \cos \theta_4\right) +
\right.
\]
\[ + r_{21} \left(\sin \theta_1 \left(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3\right) \sin \theta_4 + \cos \theta_1 \cos \theta_4\right) -
\]
\[ - r_{31} \left(\cos \theta_2 \sin \theta_3 - \sin \theta_2 \cos \theta_3\right) \sin \theta_4 \]  

(5.35)

\[ m = (r_{11} \left[\left(\cos \theta_1 \left(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3\right) \cos \theta_4 + \sin \theta_1 \cos \theta_4\right) \cos \theta_5 - \cos \theta_1 \left(\cos \theta_2 \sin \theta_3 - \sin \theta_2 \cos \theta_3\right) \sin \theta_5\right] - r_{21} \left[\left(\sin \theta_1 \left(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3\right) \cos \theta_4 - \cos \theta_1 \sin \theta_4\right) \cos \theta_5 -
\right.
\]
\[ - \sin \theta_1 \left(\cos \theta_2 \sin \theta_3 - \sin \theta_2 \cos \theta_3\right) \sin \theta_5\right] - r_{31} \left[\left(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3\right) \cos \theta_4 \cos \theta_5 +
\right.
\]
\[ + \left(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3\right) \sin \theta_5 \]  

(5.36)

The robot’s joint configuration signifies that the end-effector may be placed on the opposite side so that up to eight solutions can be carried out. The opposite configuration is obtained from equation (5.37)

\[ \theta_4' = \theta_4 + 180^\circ, \quad \theta_5' = \theta_5, \quad \theta_6' = \theta_6 + 180^\circ \]  

(5.37)

5.3.4 Path planning

Path planning concerns the method used to compute a trajectory in a 6th dimension space (which corresponds to the six joint variables \( \Theta = (\theta_1, \ldots, \theta_6) \)) from the initial to the goal position and orientation of the end-effector. In our case, the path refers to the position, velocity and acceleration of each joint during the time taken in a complete grasping action.

Path specification is usually carried out as motions of the tool frame \( (R_t) \) from its current value relative to the base robot frame \( (R_b) \). This motion, in general, involves both a change of orientation and a change in position of the tool. In addition to spatial changes in the motion, temporal specifications can also be requested.

There are many choices with which to define smooth functions on time that obey the path requirements imposed by the user. The default control motion configuration used in this thesis
corresponds to cubic polynomial functions. The inputs are the initial and the final pose of the end-effector and the amount of time. To make a single smooth motion, at least four constraints are necessary for each point:

\[
\theta_i(t_0) = \theta_{i0}, \theta_i(t_f) = \theta_{if}, \dot{\theta}_i(t_0) = 0, \dot{\theta}_i(t_f) = 0, i = 1, \ldots, 6 \tag{5.38}
\]

where \(t_0\) and \(t_f\) are the initial and final times of the trajectory (we make \(t_0 = 0\)). These constraints can be satisfied for each joint by a cubic polynomial:

\[
\theta_i(t) = \alpha_{i0} + \alpha_{i1}t + \alpha_{i2}t^2 + \alpha_{i3}t^3 \tag{5.39}
\]

The joint velocity and acceleration are:

\[
\dot{\theta}_i(t) = \alpha_{i1} + 2\alpha_{i2}t + 3\alpha_{i3}t^2 \\
\ddot{\theta}_i(t) = 2\alpha_{i2} + 6\alpha_{i3}t \tag{5.40}
\]

Upon combining equation (5.38) with equations (5.39) and (5.40) we obtain

\[
\alpha_{i0} = \alpha_{i0} \\
\alpha_{i1} = 0 \\
\alpha_{i2} = \frac{3}{t_f^2}(\theta_{if} - \theta_{i0}) \\
\alpha_{i3} = \frac{3}{t_f^3}(\theta_{if} - \theta_{i0}) \tag{5.41}
\]

and equation (5.39) eventually generates the trajectory for each joint.

5.4 Experimental test

The proposed grasping system focuses on industrial applications, in which the environmental conditions (background, lights) and robot movements are controlled. The system consists of a vision system using a monocular camera and a gripper with two flexible fingers. Figure 5.5 shows the gripper designed.

The experiments were carried out with a 6 DOF Staüblì RX-90 robot manipulator. The manipulator’s maximum payload is 11 [kg]. The CS7-M controller is used to communicate
5.4 Experimental test

Figure 5.5: Detail of the two-finger gripper and the camera integrated in the end-effector.

with the robotic arm. The gripper is mounted on the end-effector of the robot. The camera is a Philips webcam, model SPC350, and this is also integrated into the end-effector. More information about the gripper performance and setup parameters can be found in [134].

The objects to be manipulated were randomly located on a surface. The application goal is to identify the scene object, calculate its pose with regard to the robot reference system, pick up the object and place it in a box. Figure 5.6 shows the experimental setup.

Figure 5.6: Photograph of the experimental setup.

The test was carried out on eight objects from our database, and four random poses were
5. A GRASPING EXPERIENCE WITH THE CSS-NSSD APPROACH

taken for each of them. Table 5.2 shows the collection of objects used in different poses. Three factors have been taken into consideration when performing a qualitative evaluation of this grasping experience: object identification, pose estimation and grasping execution. Table 5.3 shows the results according to the success of these three factors.

In general, the system worked quite well for all the objects and poses tested. The success percentage per stage was: identification 93.7% (30/32), pose 90.6% (29/32) and grasping execution 81.1% (26/32). The errors in Table 5.3 came about for different reasons. For example, in the case of the spinning top, the identification and the pose stages took place, but the grasping execution was not successful on two occasions. This is owing to the fact that the safety contact region near the tip of the spinning top is very small and the grasp might consequently have failed there. The same problem can be seen for objects 1 and 7. In the case of object 2, the pose estimation fails but the shape’s smooth surface allows the system to accomplish a successful grasp. Note that the recognition system was able to differentiate between the lemon and the spinning objects, which are very similar shapes.

Figure 5.7 presents several frames obtained during the grasping process. Frames a), b) and c) correspond to the 3D recognition stage in which the camera is placed in three different positions until the object is recognized. In frames d) and e) the gripper is approaching the object according to the path planning solution. Frame f) corresponds to the grasp action, and in frames g) and h) the robot is carrying the object to its final position, also following the computed path. The evaluation of our grasping system must consider three factors: the object identification, the pose estimation and the grasping execution. Table 5.3 shows the results regarding the success achieved during the object identification, the pose estimation and the grasping execution.

Upon analyzing the grasping results of Table 5.3, it will be noted that the spinning object (object 2) was subject to errors during the grasping process, despite the fact that the identification and the pose estimation were considered to be successful. This is because the spinning object has a shape with a high curvature and the safety contact region is very small. Small errors in the contact point alignment process thus make the object slip from the fingers.
5.4 Experimental test

<table>
<thead>
<tr>
<th></th>
<th>Pose a</th>
<th>Pose b</th>
<th>Pose c</th>
<th>Pose d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object1</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>Object2</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>Object3</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>Object4</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
<tr>
<td>Object5</td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
</tr>
<tr>
<td>Object6</td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
</tr>
<tr>
<td>Object7</td>
<td><img src="image25.png" alt="Image" /></td>
<td><img src="image26.png" alt="Image" /></td>
<td><img src="image27.png" alt="Image" /></td>
<td><img src="image28.png" alt="Image" /></td>
</tr>
<tr>
<td>Object8</td>
<td><img src="image29.png" alt="Image" /></td>
<td><img src="image30.png" alt="Image" /></td>
<td><img src="image31.png" alt="Image" /></td>
<td><img src="image32.png" alt="Image" /></td>
</tr>
</tbody>
</table>

**Table 5.2**: Objects pose
### Table 5.3: Summary of results

<table>
<thead>
<tr>
<th>Identification</th>
<th>Pose</th>
<th>Grasping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object 1-a</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 1-b</td>
<td>OK</td>
<td>ERROR</td>
</tr>
<tr>
<td>Object 1-c</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 1-d</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 2-a</td>
<td>OK</td>
<td>ERROR</td>
</tr>
<tr>
<td>Object 2-b</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 2-c</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 2-d</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 3-a</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 3-b</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 3-c</td>
<td>OK</td>
<td>ERROR</td>
</tr>
<tr>
<td>Object 3-d</td>
<td>OK</td>
<td>ERROR</td>
</tr>
<tr>
<td>Object 4-a</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 4-b</td>
<td>ERROR</td>
<td>ERROR</td>
</tr>
<tr>
<td>Object 4-c</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 4-d</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 5-a</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 5-b</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 5-c</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 5-d</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 6-a</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 6-b</td>
<td>ERROR</td>
<td>ERROR</td>
</tr>
<tr>
<td>Object 6-c</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 6-d</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 7-a</td>
<td>OK</td>
<td>ERROR</td>
</tr>
<tr>
<td>Object 7-b</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 7-c</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 7-d</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 8-a</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 8-b</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 8-c</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Object 8-d</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>
5.4 Experimental test

Figure 5.7: Sequence of robot movements during an object manipulation task. The manipulation system is based on an active recognition system based on the proposed framework.
5. A GRASPING EXPERIENCE WITH THE CSS-NSSD APPROACH
6

Conclusions and Future Works

6.1 Contributions

This thesis is framed in the area of computer vision, and is particularly focused on the 3D recognition problem. As was discussed in Section 2, a multitude of potential solutions can be found in literature depending on several aspects such as: the kind of sensor (one or more than one camera, range finder, laser scanner, ...), the kind of scene (static/moving scene, single/complex scene, ...) and the kind of application in which the vision algorithm is going to be integrated/used (manufacturing, quality control, robot manipulation, mobile robots, ...). Of all these options, this thesis offers a 3D recognition solution providing the identification plus positioning of an isolated object from images taken by a camera on board a manipulator robot. In this framework, the 3D recognition task can therefore be converted into the integration of particular 2D shape recognition results from several camera positions, which is denoted as an active recognition system. The main contributions of this thesis can be summarized as follows.

1. An in depth qualitative and quantitative analysis with regard to the performance of 2D shape recognition methods when they are used to solve 3D object recognition problems has been carried out. To the best of our knowledge, no comparative study of different 2D shape recognition algorithms adapted to view-based 3D recognition systems has been reported until now. With regard to this point, the following aspects should be highlighted:

   (a) Well known shape descriptors (contour and regions) and 2D similarities measurements (deterministic and stochastic) have been combined to evaluate a wide range of solutions.
(b) In order to quantify the efficiency of each approach, three original parameters: Hard Recognition Rate \((Hr)\), Weak Recognition Rate \((Wr)\) and Ambiguous Recognition Rate \((Ar)\) have been proposed. These parameters open the evaluation to active recognition methods which deal with uncertainty.

(c) Up to 42 combined methods have been tested on two different experimental platforms using public database models. A detailed report of the results and a discussion, including detailed remarks and recommendations, are presented at the end of Section 2.

2. A complete framework that allows the implementation of active recognition systems based on monocular vision is proposed. This framework is independent of the type of shape descriptors (contour or region based), and the similarity measure used to develop the shape recognition process. Worthwhile aspects concerning the framework are the following:

(a) This complete framework will allow researchers to tackle the ambiguity problem for any generic object recognition system based on active recognition paradigms.

(b) Although most methods deal with uncertainty by using stochastic models, this thesis develops a mathematical background to implement different non stochastic methodologies with the goal of reducing uncertainty.

(c) An in-depth comparative analysis of: (i) different models with which to implement the proposed framework, and (ii) object recognition performance using other frameworks taken from the scientific bibliography, has been carried out. Experiments have shown that our framework exhibits a better performance than others.

(d) The experimentation of the framework was performed with a 6 DOF manipulator robot carrying a camera on its end-effector. Experimental results show the feasibility and effectiveness of our framework when used in industrial applications, since it is easy to implement, achieves high recognition rates and high computational efficiency, is robust to variations in illumination, does not require a training stage, and estimates object pose with accuracy.

3. Within the vision-robot framework proposed, a new active recognition system is defined. The system is named CSS-NSSD since the representation model is based on two main
6.2 Contrast of Results

concepts: Canonical Sphere Section (CSS) and Normalized Silhouette in the Fourier Spectral Domain (NSSD). Relevant aspects of this proposal are:

(a) The active recognition algorithm is faster and more accurate owing to the normalization and reductions of the object representation model. The Canonical Sphere Section (CSS) makes it possible to highly reduce a view-based object database when objects with reflective symmetry properties are present in the database. Moreover, NSSD makes the system robust to small 2D shape deformations and geometric transformations.

(b) The good performance of the CSS-NSSD model has been experimentally proved in two tests. Test 1 proves the performance of NSSD with other popular shape descriptors, and Test 2 compares the active recognition system with seven active recognition systems. It has been proved that the CSS-NSSD strategy leads to efficient, accurate, computationally low cost active recognition systems.

4. The vision system, containing the proposed active recognition procedure, has successfully been integrated into a robotic platform with the aim of carrying out manipulation experiences using a two finger gripper. The CSS-NSSD model efficiently provided the pose of the object and the contact points coordinates, which were the first of the grasping stages.

6.2 Contrast of Results

The results obtained during the investigation have been published and presented in various forums, some of which are presented below

Chapter 2


Chapter 3

Elizabeth González, Antonio Adán, Vicente Feliú Batlle: Framework heurístico para la implementación de sistemas activos de reconocimiento de objetos. RIAI (Submitted)
6. CONCLUSIONS AND FUTURE WORKS


Chapter 4


6.3 Future Improvements and Developments.

A few aspects of the method proposed in thesis could be improved and extended in future versions. One improvement will be addressed by simplifying the D-Sphere structure in order to make the active recognition method simpler and faster. We are aware that most of the ambiguity problems are owing to symmetries in the object model. Using all the nodes of the tessellated sphere implies storing redundant views and information in the representation model. This problem has been partially solved by using the CSS concept which is based on the reflective symmetry. However, most objects do not have this kind of symmetry. This is therefore an open issue which could be dealt with in the future.

Another clear extension of the method will be addressed by dealing with occlusion situations. The recognition solution included in this thesis assumes that the object is viewed in its entirety from any viewpoint, and that occlusion is not therefore permitted. However, in a real environment which is not strictly controlled, other objects or events might temporarily occlude the scene. The recognition system should be capable of providing an intelligent response in these cases.
Finally, the system could be extended to more difficult scenes. For example, the recognition problem in scenes with multiple objects is a challenging problem which is currently being confronted by researchers through the use of different strategies and different kinds of sensors. In this case, the active recognition system would probably solve earlier processing problems such as segmentation and occlusion. This would involve a serious revision of the 3D recognition method proposed herein.
6. CONCLUSIONS AND FUTURE WORKS
Bibliography


BIBLIOGRAPHY


