Function-based object recognition

Michael Pechuk
Function-based object recognition

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Michael Pechuk

Submitted to the Senate of the Technion - Israel Institute of Technology
Sivan 5767 HAIFA June 2007

This research thesis was done under the supervision of Professor Ehud Rivlin in the Faculty of Computer Science.

The generous financial help of the Technion is gratefully acknowledged.
Contents

1 Introduction 3

2 Related Work 7

3 Function Based Classification via Generic and Symbolic Models 19
  3.1 Segmentation 21
  3.2 Multi-Level Hierarchy Functional Structure 23
  3.3 Implementation of a Multi-Level Hierarchy 27
  3.4 Operational Multi-Level Hierarchy 32
  3.5 Learning Functionalities 34
  3.6 Classification 38
    3.6.1 Matching Grade Computation 38
    3.6.2 Matching Implementations of Multi-Level Hierarchies to Functionalities 42

4 Experimental Results 46
  4.1 Classification Performance 47
  4.2 Cross-Validation 50
  4.3 Receiver Operating Characteristic for Accuracy 50
    4.3.1 ROC: Accuracy on Synthetic Data 52
    4.3.2 ROC: Accuracy on Real Range Image Data 54
  4.4 Cluttered Scenes 54
    4.4.1 Experiments on Synthetic Data 54
    4.4.2 Experiments on Real Range Image Data 55
  4.5 Classifying Compound Objects 56
  4.6 Implementation Details 63

5 Conclusions 67

A Geometric Properties Used in Classification 71
List of Figures

2.1 Example of a Parameterized Geometric Model. 7
2.2 A simplified example of encoding object using Geon Theory representation. Objects are encoded as a four-digit code, the first digit of which encodes whether the edges are straight or curved; the second digit encodes whether the object has reflection symmetry, rotation symmetry, or both; the third digit encodes how the shape of the object changes with distance along its central axis, and the fourth digit encodes whether that central axis is straight or curved. Relations between objects can be encoded in similar manner, allowing a decomposition of objects into their component pieces, resulting in a compressed code for each complex object. 8
2.3 Instances of Aircraft in the Same Generic Family Drawn by ACRONYM. 10
2.4 History of the GRUFF system approach. 12
2.5 Identifying functional parts in a chair instance. 16
3.1 Hammer segmentation example. 22
3.2 Airplanes segmentation example. 22
3.3 An implementation of the multi-level hierarchy of functionalities that describe an ”Armchair”. 23
3.4 Associations, connections, and mappings shown on an implementation of a multi-level hierarchy of a chair. These three types of relationships are represented by bold, dashed, and dashed-and-dotted lines respectively. 24
3.5 A clique in the multi-level hierarchy functional structure. The clique corresponds to the group of functional parts Back Support, Sittable, and Ground Support, enclosed in a dashed contour boundary in Figure 3.3. These functional parts of the functional part chair are siblings, where the functional part ”Chair” represents their common ancestor. While the functional parts are represented by nodes, the relationships between each pair of them are represented by edges. In the functional hierarchy, each sibling group has a clique structure and each pair of functional or primitive parts is characterized by a relationship expressed in terms of geometric properties. 25
3.6 $imlh$ (Ground Support) is the implementation of the sub-functional part of the armchair $imlh$ described in Figure 3.3. $imlh$ (Ground Support) is the $imlh$ of a part of the entire $imlh$ (Armchair). 28
3.7 An example of imlh(A (Sittable, Ground Support)). Here, imlh(A (Sittable, Ground Support)) is the imlh of a part of the entire imlh (Armchair) and A(x,y) refers to the associations between the functional parts x and y. 29

3.8 An operational multi-level hierarchy of a chair omlh(Chair). The nodes of omlh(Chair) are sub-functional parts of the object Chair, while the signatures are sets of likelihood functions built from histograms. These functions provide grades for matching for different parts of the entire layered functional structure. 33

3.9 Construction of several imlh’s, an omlh, and their use in the case of a chair. Each image in the testing set is segmented and an imlh is constructed for it. These imlh’s are input to the learning stage, in which we compute an omlh. At classification, the analyzed image is segmented to primitive parts and the grade of matching to any learned class is evaluated by means of the omlh. 35

3.10 Learning and classification flows. 37

3.11 The search state DAG. This search DAG has a depth of three. 43

4.1 Example of spectacles classification. (a) The image of the object. (b) The primitive parts detected after segmentation. (c) Result of classification, the resulting functional parts are shown with different colors. 47

4.2 Learning real spoons and classifying real spoons and spectacles. The average grades of the spoons and the spectacles are separated by 0.1. This separation assures very accurate classification. 48

4.3 Testing classification accuracy. (a) represents a subset of the training set images. (b) illustrates three-legged chairs used in the experiment. All of them were correctly classified. (c) and (d) illustrate one-legged chairs that were used in the experiment. While the chairs in (c) were correctly classified, the scheme could not cope with the object in (d), due to its unusual back-support implemented by a blob. 49

4.4 Cross-validation on the whole data base, which includes spectacles, forks, spoons, mugs, stools, tables, chairs, and airplanes. 51

4.5 Experiments on synthetic data. Synthetic 3D models of mugs and stools are employed. In this experiment, we considered a classifier that selects the maximum grades for the different components. We present the classification grade threshold on the abscissa. (a) The classifier’s accuracy of mugs versus mugs and stools. (b) The mug classifier’s accuracy versus that of mugs and stools. (c) Overall accuracy. 53

4.6 Experiments on real range image data. We show the classification grade threshold on the abscissa. (a) A receiver operating characteristic on the whole data base for stools (the uppermost curve – dashed) forks (the middle curve - normal ), and spectacles (the lowest curve - dot and dashed). (b) The classifier’s accuracy of stools versus other objects in the database. (c) Overall accuracy. 55
4.7 A cluttered scene of a chair and a synthetically enlarged fork (left). The chair is correctly classified (right). ........................................... 56

4.8 Two cluttered scenes of a chair and a table in a room. (a) and (b) represent two examples of the first type of image. Figure (c) and (d) represent the results of segmentation of the scenes in (a) and (b), respectively. The primitive parts sticks and plates are shown as they are detected and modelled in the segmentation phase. Figure (e) and (f) represent the results of classifying the scenes in (a) and (b), respectively. (e) and (f) show the resulting functional parts – the back support, seat, and the ground support – with different textures. ................................................................. 57

4.9 Images of some chairs used in the experiments. ................................. 58

4.10 Images of some forks and spoons used in our experiments. ................. 59

4.11 Images of several airplanes ............................................................ 60

4.12 Images of several cupboards .......................................................... 61

4.13 Images of some spectacles used in the experiments. ......................... 62

4.14 Illustration of an mlh(Bedroom) representation with height four. ......... 63

4.15 Examples of two bedrooms and two dining rooms we used in experiments. 64

4.16 Classification of a bedroom. By definition, a bedroom includes at least a bed. In this example, we show the classification of bedrooms versus dining rooms. (a) represents a digital photo of a room with a chair and a bed. The magenta color represents unselected parts during the classification process. The image in (b) represents the classification of the room as a bedroom, after the chair and the bed are selected (and colored) for analysis. ......................... 65

4.17 Illustration of reusable function based learning. ............................... 65

5.1 Scissors example. The holes in graspers are highlighted by red ellipses. ... 69
List of Tables

A.1 Geometric properties for primitive and functional parts. . . . . . . . . . . . . 72
A.2 Geometric properties for connections. . . . . . . . . . . . . . . . . . . . . . . . 73
We propose a novel scheme for using supervised learning for function based classification of objects in 3D images. During the learning process, a generic multi-level hierarchical description of object classes is constructed. The object classes are described in terms of functional components. The multi-level hierarchy is designed and constructed using a large set of signature-based reasoning and grading mechanism. This set employs likelihood functions that are built as radial-based functions from the histograms of the object instances. During classification, a probabilistic matching measure is used to search through a finite graph to find the best assignment of geometric parts to the functional structures of each class. An object is assigned to a class providing the highest matching value. Reuse of functional primitives in different classes enables easy expansion to new categories. We tested the proposed scheme on a database of about one thousand different 3D objects. The proposed scheme achieved high classification accuracy while using small training sets.
Symbols

ROC  receiver operator characteristic
$K$  the Gaussian curvature
$H$  the mean curvature
Chapter 1

Introduction

The problem of object classification from sensory data is defined, in the literature, as the association of visual input to a name or a symbol. Although much research on the topic has been published, the community still lacks usable vision systems that can classify a large number of objects (natural or man-made). Early work in this area focused on the use of explicit geometric models for object recognition. Numerous surveys give examples of such research efforts in “model based” and “CAD based” vision (e.g., see [3, 6, 46]. However, for any reasonably complex environment, such approaches are ultimately limited by either the storage space required for the set of models or the processing speed at which matching can be performed. Even given sufficient computing resources, it is unlikely that all the parameterizations or component descriptions necessary to sufficiently generalize a prototype object model to an entire object category could readily be anticipated.

Therefore, while traditional model based vision has been and still is a useful research paradigm, there is an increasing awareness that something more will be needed in order to achieve visual perception for “autonomous”, “real world” systems. The system must be
able to recognize and deal with truly novel objects - that is, objects for which it has no explicit prior model, at least in the traditional sense of the word “model”. This trend has led to computer vision research which emphasizes the development and use of generic object models. A “generic” model is meant to capture a category of objects, as opposed to one particular instance. Part of the inspiration for this approach has come from the field of psychology (see [21, 8]), where researchers such as Rosch have done extensive work on how humans develop category concepts [39, 45, 44, 46]. Categorization by humans is attributed to the idea that all possible combinations of attributes do not occur in the physical world. Therefore, humans develop an organization of the world in terms of hierarchical category concepts, grouping objects with similar characteristics. In Rosch’s hierarchy, there are three major levels of specificity. For example, a basic or entry level category is the name most commonly given to an object, such as “cup”. To represent a refinement or specialization of a basic category there are subordinate categories, like “mug” or “juice glass“. Finally, a generalization of a basic category would be a superordinate category description, such as “dishes”.

Researchers in artificial intelligence were the first to begin incorporating ideas about object classes into recognition systems.

Well known early work in the area was done by Winston et al. [57]. They pointed out that there can be an in finite number of different shape descriptions for objects in a category as simple as a cup, but that a single functional description can be used to represent all cups in a concise manner. As researchers in these fields and computer vision have gained a greater
understanding of the complexity of object recognition, the idea of using functional characteristics of an object as an aid to recognition has further developed. In brief, functional analysis of an object relies on analyzing the object to determine if it can satisfy the requirements of some category of objects or tasks. In this way, objects which are truly “novel”, or which contain a unique configuration of parts which has never been seen before, can still be categorized as being able to potentially function as a member of a particular category of objects.

The goal of this thesis is to demonstrate a new, function-based object recognition scheme that is able to classify objects from range images. Our novel scheme uses learning for function-based classification of objects from 3D images. The classification process calls for constructing a generic multi-level hierarchical description of object classes. The object classes are described in terms of functional components. The multi-level hierarchy provides a nesting mechanism for functional parts and has unbounded depth. In this context, the construction of the generic multi-level hierarchy can be thought of as a learning phase.

In the learning phase, the input range data describing each object instance is segmented, each object part is labeled as one of a few possible primitives, and each group of primitive parts is tagged by a functional symbol. Connections between primitive parts are also computed in the segmentation stage. We refer to the input instances as implementations of the multi-level hierarchy that defines a class. We then define a classification scheme using histograms built from the observed functionalities of a number of object instances. This scheme is a probabilistic model of an object class and we call it the operational multi-level
hierarchy. Our scheme can automatically build the description of any functional describable object class from labeled examples.

Function-based approaches offer the advantage of reusable learning: functional parts that have the same purpose and are shared between different classes do not have to be relearned with each new class, which notably accelerates the learning phase.

We tested the proposed scheme on a database of about one thousand different 3D objects. No other classification (or recognition) scheme has yet been tested on hundreds of real objects captured in range images.

The structure of this work is as follows. In Chapter 2 we present an overview of the literature on function based recognition from 3D data. Next, in Chapter 3, we describe the proposed method. In this section we present the details of segmentation (Section 3.1), the multi-level hierarchy functional structure (Section 3.2), supervised learning (Section 3.5), and the proposed classification method (Section 3.6). In Chapter 4, we show our experimental results. Next, we conclude in Chapter 5. Finally, in appendix A we present the geometric properties used in the classification scheme.
Chapter 2
Related Work

Most of the work addressing the 3D recognition problem uses a *model-based* approach, where input is matched to models of objects. Several researchers use a *geometrical model* in which the input is directly matched to a model of low-level geometrical features; see works of Dorai and Jain [17], Fan et al. [18], Hebert et al. [24] Hoffman et al. [26] and Vemuri et al. [56]. Most of these works are confined to specific domains of object shapes and are not robust

![Parameterized Geometric Model](image)

Figure 2.1: Example of a Parameterized Geometric Model.
enough to partial 3-D knowledge and view-dependent input. Moreover, these approaches are not easily applicable for more generic recognition, where no precise geometrical models exist. Later on, the parameterized geometric modelling was introduced by Ullman [54]. A parameterized geometric model is created by replacing some of the constants in a geometric model with parameters that may be constrained to lie within defined ranges (see Figure 2.1).

The structural model is more high-level and cognitive-based: input objects are recognized.
by matching their parts and the connections between them to the model. The cognitive basis for this approach was set by Biederman [8] in his RBC (recognition-by-components) concept. Biederman notes that certain properties of visual features remain invariant to perspective transformation through small angles. For example a straight edge appears straight, while a curved edge appears curved, through a wide range of rotations of the object, although the exact angle or curvature of that edge changes with rotation. Biederman thus proposes the Geon Theory, which suggest that an object category in human recognition is defined as a particular set of qualitatively-described primitive 3-D shapes called geons and the qualitative relationships between them. A simplified example of this concept is illustrated in Figure 2.2. For example the teapot shown in Figure 2.2 would be encoded by its three component objects (handle, body, spout) and the relations between them as:

- 1301: straight-edged, reflection and rotation symmetric, constant sweep, curved central axis;
- 037: smaller than, beside, join both ends to side;
- 1310: straight-edged, reflection and rotation symmetric, expanding sweep, straight central axis;
- 136: bigger than, beside, join side to end;
- 1321: straight-edged, reflection and rotation symmetric, contracting sweep, curved central axis;

These properties would remain relatively invariant through small rotations of the object.
Recognition is based on indexing the structural composition of the object. Examples for such an approach can be found in Bergevin et al. [5], Kriegman et al. [33] and Pentland [40].

As noted earlier, one motivation for using a structural model is to make it convenient to specify sophisticated parameterized geometry. However, it also becomes possible to parameterize the structure itself. In a parameterized structural model, variations in the essential structure of the model may be modeled by parameterizing the number and location of some of the component parts of the object.

One of the earliest uses of parameterized structural models occurs in Brooks’ ACRONYM system [10, 11], where families of aircraft are represented by a composition of a variable number of parts (see Figure 2.3).

Although these approaches are more generic and robust than the geometric model based schemes, they are still confined to recognition problems in which the shape of the objects to be recognized is known. Note that instances of even simple classes of objects differ from one
another greatly in terms of shape and structure.

The difficulties described for the identification of objects are equally relevant to classification. Categorization of objects, unlike identification, involves a higher level of reasoning and understanding of the object’s purpose. This high-level reasoning is not related directly to shape: instances of the same class often look very different. The imaged objects are not actually known to the classifier; thus, so any straightforward matching of the input to a known database, feasible in identification, is not applicable here. Therefore, one should obtain a set of high-level criteria and properties which are distinct and general enough to describe a class of objects, as well as a means of extracting such properties from the input.

The need for “true” generic models for representing classes of objects for classification has given rise to functional model approaches. An approach fundamentally different from the previous ones was introduced by Gibson [21], who considered that the human mind classifies objects according to their use, that is, by the functions they might fulfil. Winston et al. [57] propose a theory explaining how physical models are learned and identified using functional definitions, physical examples, and precedents. DiManzo et al. [16] present the FUR (FUnctional Reasoning) project, a functional reasoning and shape-function integration system, in which several primitive functions are presented (support, grasp, enter, hang, etc.), and several functional expert concepts for identification of functional primitives are discussed.

An impressive number of results in the function-based classification field were demonstrated in seminal work of Bower and Stark with the GRUFF and OMLET systems (see
Figure 2.4. The Bowyer et al. in [48, 50], propose concepts for function-based recognition of multiple object categories in their GRUFF system. Their paper addresses the reuse of a limited set of knowledge primitives for defining an expanded domain of competence and computing an association measure for the appropriateness of a shape that can be compared across categories. This method allows different interpretations of a shape to be rank ordered. The authors also discuss the categorization of several basic-level categories, performed on a simple polyhedral boundary representation (a CAD-like model), as input. Several functional knowledge primitives were used. Also covered in this paper is the function-based definition of a category, specified by a set of functional properties, where each functional property is
implemented as a set of invocations of the knowledge primitives.

An efficient indexing of the knowledge base is proposed where during the recognition process an accumulated association measure reflects the systems confidence in support of the shape belonging to the hypothesized category. In the GRUFF-3 system (Sutton et al. [50]) the concepts of the GRUFF system are applied on several sub-categories of the superordinate category of dishes (supported sub-categories are: cup/glass, plate, pot/pan, bowl, pitcher). The input to the system, as in the GRUFF-2, is a simple polyhedral boundary representation. Towards a categorization of dishes, one more knowledge primitive is proposed (in addition to the five primitives defined in GRUFF-2): enclosure (testing if there exists a concavity in the shape which can be closed by a single plane).

Green et al. [22] describe a function-based recognition system for dealing with objects whose function depends on parts joined by articulated connections (focusing on the scissors sub-category). The suggested method consists of two stages: the first stage is the recovering of an articulated shape model from a sequence of simple 3-D shape descriptions. This model consists of individual boundary representation for the parts, connections between them allowing rotational or translational motion between two parts and linkage relations representing causal effects between two connections. The second stage is an analysis in which the functional potential of the articulated model is determined, following a GRUFF category tree.

The first attempts at categorizing from incomplete 3-D shape object descriptions were made in Stark et al. [49], where the authors describe an application of the GRUFF concepts
of function-based reasoning to an OPUS model (object plus unseen space). More recently, Sutton et al. [51] built a testing framework for the GRUFF system using stereo vision images. However, recognition accuracy is not detailed. Note that GRUFF was developed for the learning of membership functions from 3D objects Woods et al. [58]. The goal of the learning phase is to augment functions that are defined in a human-driven preprocessing stage. GRUFF and OMLET were extensively tested on raw images that included chairs artificially constructed from boxes.

The possible advantages of functional approaches for generic classification were recognized in several relatively early works, such as DiManzo et al. [16] and Winston et al. [57]. Following these concepts, several systems for object classification were built (see Bajcsy et al.[4], DiManzo et al. [16], Stark et al. [48], and Sutton et al. [50]). However, little experimental work has been done to test these concepts. Only preliminary attempts were made towards functional classification from raw images of objects Stark et al.[49] and stereo vision based models Sutton et al.[51]. Existing models for applying function to representation for the purpose of classification still fail to present true generic models. Nor do they present robust methods of relating robust high-level concepts of functional representation to low-level images.

Parallel to the analysis of static 3D models is a promising new trend in which sequences of images are used to understand human interactions with the environment. The authors in Peursum et al. [41], proposed a scheme for analyzing sequences of images in which a person interacts with a room of chairs. While the images are being segmented, the elements in the
room are labelled by measuring the degree with which the person interacts with them. The system tracks the human operator.

Verification, – measuring the degree at which recognition is performed – is important for evaluating artificial intelligence and computer vision systems. Comparing implemented schemes to human abilities Turing [53] is important as well. Following Turing [53], the authors of Amant et al. [47] provide an overview of research on the use of physical tools and claim that intelligence can be evaluated by analyzing the activity of agents as tools users.

Li and Lee [34] propose a system for recognizing articulated objects. This system employs a learning stage that is based on accumulative Hopfield matching, also known as attributed relational graph matching.

Bajcsy and Solina [4] propose a modelling system for generic objects. It consists of a prototype made up of parts (each part is a superquadric) and is based on the psychological notions of categorization suggested by Rosch [44, 45]. Rivlin et al. [42] describe a framework for recognition by functional parts using a combination of functional primitives, volumetric shape primitives, and their relationships. First, the input images are segmented into parts, which are further fitted to deformable superquadrics. Each part is classified into one of four types (strips, sticks, blobs, and plates). The recognition process is based on finding functional features embedded in relationships and attachments between pairs of parts (as well as other shape features).

Vaina and Jaulent. [55] propose a model for recognizing functionalities, combining representations of shapes and object categories with goal requirements for actions. In this context,
additional high-level functional concepts are presented by Hodges [25], where the authors describe the problem of improvisation. The authors study the relationship between physical properties of objects, their functional and behavioral representation, and their use in problem solving. The key concept in this approach is that functional classification organizes the information about objects into taxonomies varying in the degree of compatibility between action-requirements and object properties. According to the proposed model, Object representation is constructed of three modules: the object category module (being a hierarchical representation of objects in terms of their semantic categories), the object-concept module (where parts, their relations and functions are made explicit) and object-structure module (in which the shape of parts, their geometrical relations and visual attributes are addressed). The properties necessary and sufficient for achieving the goal of an action are collectively...
called action-requirements. A model of compatibility between objects and actions is introduced and is demonstrated on the domain of hand-actions and relevant objects.

Kise et al. [30, 31] propose a model for functional object recognition supporting dynamic functions (such as magnifying force, as is the case in hand tools) in addition to the static functions (such as support) introduced in previous object recognition models. In order to support these dynamic functions, the paper introduces knowledge about mechanisms (in terms of basic machines such as levers), and about actions (how the object is to be used). The proposed recognition procedure using this functional model associates this knowledge to found structure and shape. The proposed system is demonstrated on the recognition of several bottle-openers, screw-drivers and wrenches, where the input objects are given in the form of a full polyhedral model.

More recently, Keselman and Dickinson. [29] propose a generic model in which inferences are drawn from examples. The model is guaranteed to be generic because it entails a high level representation of the input object, segmented and partitioned into adjacency-related regions. The regions are groups of 2D information elements that represent the lowest common abstraction detectable in the sequences of input images. Although their model is generic, the authors propose its implementation in direct recognition tasks as future research. Moreover, the system assumes that the viewing direction of different input examples is consistent, and the problem of merging additional views is proposed as a future research topic as well.

Brand [9] describes a physics based application for automatic understanding of gear mechanisms. Functional inferences are employed for reverse engineering, thus making it
possible to cope with occlusion and partial information. Rivlin et al. [43] presented a theory of function-based recognition that is a natural extension of the part-based shape recognition approach. Following Rivlin et al. [43], Froimovich et al. [20] propose a system for function-based classification. The classification approach, performed on range images of real 3D objects, assumes a priori knowledge of the objects (See Figure 2.5 for example of partitioning into functional parts, as identified on a chair).

A good overview of function based classification methods can be found in Bicici et al. [7]. The authors feel that common sense reasoning is a particular, specialized, and very high level kind of functional reasoning (e.g. Davis [14], Minsky [36] and Morgenstern [37]).
Chapter 3

Function Based Classification via Generic and Symbolic Models

Our proposed scheme consists of two phases: supervised learning and classification. Each of these phases receives as input segmented images. The objects are segmented into primitive parts. For learning, the segmented parts are also grouped and labeled into functional parts.

Following Rivlin et al. [43], the primitive parts that we consider are sticks, plates, and blobs, where the first two can be deformed. A functional part is defined as an object part that can provide a certain functionality and comprises several primitive parts; for example, the ground support of a chair might consist of four parallel stick primitive parts. It is stated in Biederman [8] that thousands of objects can be mapped to a structure consisting of only a few primitive parts. The immense number of objects in nature is the result of the combinatorial number of interrelationships between the primitive parts.

We describe functionalities of objects in terms of multi-level hierarchies. In this context, we consider that each functionality can be decomposed into sub-functionalities. We map functionalities and sub-functionalities to nodes and their children respectively, in tree-like
structures (see Section 3.2 and Figure 3.3). We use multi-level hierarchies in learning as well as in classification.

In the learning phase, several instances (objects) of a class are input. We implicitly provide mappings of primitive parts (to functional ones) in the input and employ supervision in learning. In the learning phase, we compute the values of the geometric properties of the constituents and the relationships between them. We refer to the segmented and labeled images as implementations of the multi-level hierarchy of functionalities that describe the input objects.

In the learning phase several implementations of multi-level hierarchies are received as input and an operational multi-level hierarchy is created. This hierarchy will be defined in Section 3.4 (see Figure 3.8). During learning, the values of the geometric properties are merged together in signatures implemented by RBF (radial-based functions) [12, 35]. Once the learning phase is completed, the operational multi-level hierarchy, which is a generic representation, has accumulated enough information for classification – the next phase. In the classification stage, the operational multi-level hierarchy is used to provide matching grades to the input objects.

In what follows, we present the details of the segmentation process, the multi-level hierarchy functional structure, its implementation and operational use, as well as the supervised learning and the proposed classification method. Our order of presentation, follows the natural order in which components are applied.
3.1 Segmentation

The input to our classification scheme, and thus to the lowest-level processing stage, is a raw range image represented as a point cloud (a 3D point cloud from raw range images can be seen as a parameterized or a grid driven structure). The output of this phase is a segmentation of the point cloud into regions, where a region refers to a collection of image points having similar geometrical properties.

A seminal work in the comparison of algorithms designed for segmentation of raw range images is Hoover et al. [27]. This work summarizes four range segmentation algorithms and presents a comparative framework for testing and comparing them. The authors [27] conclude that the so-called UE (University of Edinburgh) algorithm, based on Gaussian and mean curvatures estimation, provides the best results.

The learning and the classification phases receive as input primitive parts as detected by the segmentation algorithm, which is a variation on the UE algorithm [27]. This algorithm is a convenient choice for our purposes because it allowed us to detect both exact and deformed primitives such as planes and deformed planes. Although segmentation is not a major issue here, it is necessary in order to provide representation models for the primitive parts. Segmentation results for a range images of hammer are shown in Figure 3.1. Segmentation results for a range images of plastic airplane models are shown in Figure 3.2.
Figure 3.1: Hammer segmentation example.

Figure 3.2: Airplanes segmentation example.
Example of a clique

Figure 3.3: An implementation of the multi-level hierarchy of functionalities that describe an "Armchair".

3.2 Multi-Level Hierarchy Functional Structure

The classification process comprises an analysis both of the detected primitive parts and the relationships among them (see Figure 3.3). Each primitive part or group of primitive parts and the connections among them that can fulfil a certain functionality are classified as a functional part [43]. This approach is known in the literature as recognition/classification by functional parts.
Figure 3.4: Associations, connections, and mappings shown on an implementation of a multi-level hierarchy of a chair. These three types of relationships are represented by bold, dashed, and dashed-and-dotted lines respectively.
Figure 3.5: A clique in the multi-level hierarchy functional structure. The clique corresponds to the group of functional parts Back Support, Sittable, and Ground Support, enclosed in a dashed contour boundary in Figure 3.3. These functional parts of the functional part chair are siblings, where the functional part "Chair" represents their common ancestor. While the functional parts are represented by nodes, the relationships between each pair of them are represented by edges. In the functional hierarchy, each sibling group has a clique structure and each pair of functional or primitive parts is characterized by a relationship expressed in terms of geometric properties.
We have generalized the mechanism of decomposition into parts and relationships into a multi-level approach in the following sense. We define three types of relationships: associations, connections, and mappings. We call the relationships between primitive parts connections. A relationship between any pair functional parts is called an association. We define a relationship between a functional part and a primitive one as a mapping relationship. We show these three types of relationships in Figure 3.4.

Note that two functional parts that form associations can be siblings sharing a common direct functional parent node, or they can have a relationship of inclusion, i.e., one can be a sub-functionality of the other. Furthermore, several functional parts and the relationships among them can define a functionality and can form a higher level functional part.

Connections are obtained at segmentation stages. Note that segmentation can provide cues about the connectivity and the occlusion among different primitive parts.

Relationships among primitive and functional parts are inclusions. That is, several primitive parts can map to a functional part, thus forming a mapping or defining a partition from primitive parts to functional ones. In this work, we employ mappings for functional parts that are insofar as possible granular. In other words, they are leaves in the functional hierarchy. For example, in Figure 3.4, Sticks 1, 2, 3, and 4 are primitive parts that map to the Ground Support functional part and not to higher functional parts (in the multi-level hierarchy). Finding mapping relationships is equivalent to computing partitions of the input image primitives to functional ones.

The proposed hierarchy can be as complex as one wishes and is needed in order to describe
the functionalities at the desired refinement level. Here, by refinement level, we mean the possibility of classifying objects with specializations. For example, we might want to classify chairs only or we might want to classify chairs that also have arm supports.

Assume \( f \) is a functional part whose decomposition into sub-functional parts is known. We will refer to this layered decomposition as a multi-level hierarchy and denote it by \( mlh(f) \) (see the upper part of Figure 3.4 for an example of a \( mlh(Chair) \)). This structure does not refer to the primitive parts of \( f \); however, it includes the symbolic functional parts and associations between adjacent ones – those that are siblings of a common parent or have a relationship of inclusion.

### 3.3 Implementation of a Multi-Level Hierarchy

In the following, the term implementation of a multi-level hierarchy of the functional part \( f \) refers to the multi-level hierarchy of \( f \), together with its primitive parts, the geometric property values of the functional and primitive parts nodes, and the relationships among the parts (see Figures 3.3 and 3.4). We denote the implementation of the multi-level hierarchy of \( f \) by \( imlh(f) \).

If \( s \) is a functional node in the \( mlh \) of \( f \), then define \( imlh(s) \) as the sub-tree like layered structure which has \( s \) as a root and is part of \( imlh \) (see Figure 3.6). If \( s \) is an association or a connection, then let \( imlh(s) \) be the edge-like sub-structure of \( imlh \) that consists of all the geometric property values that are constituents of this relationship; see Figure 3.7. Of course, \( imlh \) and \( imlh(f) \) are equal.
Figure 3.6: *imlh* (Ground Support) is the implementation of the sub-functional part of the armchair *imlh* described in Figure 3.3. *imlh* (Ground Support) is the *imlh* of a part of the entire *imlh* (Armchair).
Figure 3.7: An example of \( \text{imlh}(A \text{ (Sittable, Ground Support)}) \). Here, \( \text{imlh}(A \text{ (Sittable, Ground Support)}) \) is the \( \text{imlh} \) of a part of the entire \( \text{imlh} \) (Armchair) and \( A(x,y) \) refers to the associations between the functional parts \( x \) and \( y \).
Associations and connections are expressed in terms of geometric properties. Furthermore, siblings of functional parts sharing a common parent functional part (as a common functional part ancestor) are grouped in cliques in the functional hierarchy. Each pair of functional parts (in the clique) are characterized by a relationship expressed in terms of geometric properties. For example, an implementation of the multi-level hierarchy of functionalities that describes an “Armchair” is shown in Figure 3.3. Each pair of functional parts from “Back Support,” “Sittable,” and “Ground Support” are connected, forming a clique as shown in Figure 3.5. Note that these three nodes have the common ancestor “Chair.” In this context, different functionalities have different complexities. Furthermore, the more complex the description of a class is, the more specialized the classification is.

Cliques represent groupings of functional parts that cooperate toward realizing a higher functional task. Note that cliques can be avoided by employing multi-levels in functional hierarchies.

The multi-level hierarchy functional structure of an object class is implemented by a layered tree-like structure. Assume $f$ is a functionality and $s$ is a sub-functional part of $f$. For any such sub-functional part $s$, define $P(s)$ and $F(s)$ to be the set of immediate primitive or functional constituents of $s$, and let $C(s)$ and $A(s)$ be the set of connections and associations between the elements of $P(s)$ and $F(s)$ respectively; see Figures 3.3 and 3.5. Note that $s$ can be $f$ itself.

In Figure 3.3, the “Arm Support” represents a functional part that supports the arms.
The “Armchair” is a higher level functional part because it describes a more complex functionality. In particular, it includes the “Arm Support” sub-functionality. In this example, \( F(\text{Armchair}) = \{\text{Arm Support, Chair}\} \). Functional part siblings are organized in cliques of associations. For example, the siblings “Back Support,” “Sittable,” and “Ground Support” are grouped in a clique, indicated in the figure by a dashed contour boundary. Two additional cliques are enclosed in dashed rectangles as well.

Note that one of \( P(s) \) or \( F(s) \) is empty and the other is not empty for any sub-functionality \( s \) of \( f \). For example, \( F(\text{Sittable}) = \emptyset, P(\text{Sittable}) = \{\text{a plate}\} \) and \( F(\text{Armchair}) = \{\text{Arm Support, Chair}\}, P(\text{Armchair}) = \emptyset \). For clarity, if \( s \) is a terminal node in \( mlh(f) \), then \( P(s) \neq \emptyset, F(s) = \emptyset \), while if \( s \) is an internal functional node, then \( P(s) = \emptyset, F(s) \neq \emptyset \).

For any symbolic primitive part, functional part, connection, or association \( x \), we define \( GP(x) \) to be the set of geometric properties of \( x \). If \( x \) is a primitive or a functional part, then \( GP(x) \) includes, among other properties, inertia moments, stability, and regularity. If \( x \) is a connection, \( GP(x) \) includes, for example, occlusion or geometric connection \( (C^{(0)} \text{continuity}) \). If \( x \) is an association, \( GP(x) \) includes, among other properties, ratio of volumes and context-based stability.

The geometric property of regularity can serve as an example of the kinds of geometric properties we used for pairs of functional parts. Regularity can be planar and circular. We implemented tests for the parallelism of sticks. Sticks can be located in a plane and they can form a circular shape as well. Moreover, we define the context-based stability property of an association as the stability of the ensemble of parts related to this association. The
context-based stability of an ensemble can be computed by checking if the projection (along
the normal to the ground) of the center of the mass crosses the bounding box of one of the
direct functional part constituents of the ensemble.

We present a partial list of geometric properties evaluated for primitive and functional
parts in Table A.1 and for associations in Table A.2, both in Appendix A. The full description
of the geometric properties we have considered is relatively large and can be found in [38].

### 3.4 Operational Multi-Level Hierarchy

Consider a multi-level hierarchy and let $P$ and $F$ be the set of all the symbolic primitives
and functional parts, respectively, that the hierarchy includes. Define

$$GP_{PF} = \left\{ (x, g) \mid x \in F \bigcup P, g \in GP(x) \right\}$$

and

$$GP_{CA} = \bigcup_{f \in F} \left\{ (y, g) \mid y \in C(f) \bigcup A(f), g \in GP(y) \right\}.$$ 

Then, the multi-level hierarchy of a functionality $f$ induces a function $H_f : GP_{PF} \cup GP_{CA} \rightarrow H$, where $H = \{ h \mid h : R \rightarrow [0..1] \}$ is the set of all (normalized) histograms that can be im-
plemented as B-spline functions (see Sections 3.5 and 4.6). These histograms are normalized
to 1. Therefore, they are likelihood functions. Evaluating $H_f$ produces a histogram function.
The histogram functions $h$ translate the values of geometric properties specific to the part
at hand into a normalized probabilistic grade.

Each geometric property is associated with a histogram of measured values. For each
Figure 3.8: An operational multi-level hierarchy of a chair omlh(Chair). The nodes of omlh(Chair) are sub-functional parts of the object Chair, while the signatures are sets of likelihood functions built from histograms. These functions provide grades for matching for different parts of the entire layered functional structure.
functional part, the set of histograms of its constituents, functional (sub)-parts and associations, represents the signature of the functional part. The signatures are computed from the instances of multi-level hierarchy implementations. The signatures label the functional parts and their associations in the operational multi-level hierarchy (omlh).

We define the operational multi-level hierarchy functional part $f$ as $mlh(f)$ together with the signatures of the functional nodes and associations in $mlh(f)$. We will use the notation $omlh(f)$ for the operational multi-level hierarchy of $f$. We illustrate an operational multi-level hierarchy of a chair in Figure 3.8. Note that $omlh(f)$ only includes sub-functional parts of $f$ and associations among sub-functionalities of $f$. $omlh(f)$ does not include primitive parts. The operational multi-level hierarchies of each learnt class are stored in a database. The notation $omlh$ describes the final tool towards classification.

The notation $mlh$ describes the functionality of a class in abstract symbolic terms. During learning, our scheme receives several input objects. We construct an $imlh$ for each one of the objects in the segmentation stage. All the $imlhs$ share a common $mlh$. In the learning stage we construct for all the instances an $omlh$. We show this construction in Figure 3.9.

### 3.5 Learning Functionalities

The left-hand side of Figure 3.10 shows the flow of the learning phase of our scheme. The input of the learning phase is a set of labeled objects described by implementations of multi-level hierarchies. Each functional and primitive part is labeled with a symbol or a generic name. Examples of functional and primitive part symbols are “ground support” and “stick,” respectively. For each input object, the proposed scheme calculates the values for all the
Figure 3.9: Construction of several $imlh$’s, an $omlh$, and their use in the case of a chair. Each image in the testing set is segmented and an $imlh$ is constructed for it. These $imlh$’s are input to the learning stage, in which we compute an $omlh$. At classification, the analyzed image is segmented to primitive parts and the grade of matching to any learned class is evaluated by means of the $omlh$. 
pre-defined geometrical properties. Furthermore, these properties are subject to RBF-like (radial-based function) learning [12, 35]. The result is an operational multi-level hierarchy for each learnt object.

We chose radial-based function learning because we learn from positive examples only. When we consider that our scheme is designed for general purpose classification of 3D objects, the number of negative examples for each desired class is huge. We analyzed other techniques and chose the one that seems the most straightforward.

In the learning phase, the scheme builds histograms for geometric properties of the functional parts as well as for the associations (see Section 4.6). The continuous domain of measured values for geometric properties is approximated by discrete accumulation values which are provided as the coefficients of the B-spline functions that match the histograms. The scalar coefficients are normalized such that the maximum coefficient equals 1.0. Note that this process is automatic, and requires no operator intervention other than labeling.

Function-based approaches offer the advantage of reusable learning: functional parts that are shared between different classes do not have to be learnt with each new class. We exploit this advantage to speed up the scheme for learning new objects; that is, we design the learning sequences from objects with functional parts having different shapes. Relearning geometrically similar functional parts is unnecessary. For example, for the armchair in Figure 3.3, if we have already learned the chair sub-part, we only have to add the signatures of the arms to the multi-level hierarchy of the armchair.
Figure 3.10: Learning and classification flows.
3.6 Classification

Figure 3.10 (right-hand side) shows the flow of the classification phase of our scheme. In the classification mode, the input consists of a set of primitive parts, the connections between them, and the database of operational multi-level hierarchies provided by the learning phase. The database contains an operational multi-level hierarchy for each learnt object. The classification phase computes a vector of grades that describes how an object conforms to class functionalities. Each element of the vector represents a grade for one class.

The class with the highest matching grade is chosen as the best match. For each one of the learned classes the scheme tries to find the best multi-level hierarchy implementation out of the given set of primitive parts and the connections between them. The best match is of course subject to the maximum matching grade.

Thus, we reduce the problem of classifying a new object to the problem of finding the implementation of a multi-level hierarchy with the highest matching grade. In fact, our classification scheme relies on computing a partition of primitive parts. The following sections describe the matching grade computation process as well as the mathematic and algorithmic details of computing partitions.

3.6.1 Matching Grade Computation

Assume we want to evaluate the matching grade for functionality \( f \). Let \( mlh \) be the multi-level hierarchy of \( f \), \( imlh \) be any implementation of \( mlh \), and \( omlh \) an operational multi-level hierarchy computed for \( f \). Let \( Prim \) be an input set of primitive parts from a segmented
image. Let \( \text{imlh} \) be the set of all implementations of instances of \( mlh \) built with primitive parts from \( \text{Prim} \).

Recall that we use the notation \( \text{imlh}(s) \) for the sub-tree like layered structure which has \( s \) as a root and is part of \( \text{imlh} \), where \( s \) is a functional node in the \( mlh \) of \( f \) (see Figure 3.6). Moreover, we also use the notation \( \text{imlh}(s) \) for the (edge-like) sub-structure of \( \text{imlh} \) that consists of all the geometric property values that are constituents of this relationship, whenever \( s \) is an association or a connection; see Figure 3.7.

Let \( s \) be a node sub-functional part or an association or a connection and \( g \) a geometric property. Let \( w(s, g) \) be weight functions that are proportional with the standard deviation [28] of the histogram function which itself corresponds to \( s \) and \( g \). Let \( H_s(s, g)(\text{imlh}(s)) \) be the value of the geometric property histogram for \( s \) implemented after the implementation \( \text{imlh}(s) \). Then, define

\[
\text{f-grade}(s, \text{imlh}(s)) = \sum_{g \in \text{GP}(s)} w(s, g) H_s(s, g)(\text{imlh}(s)).
\] (3.6.1)

In this context, for any association \( r \), let \( \text{imlh}(r) \) be the sub-part of \( \text{imlh} \) that contains the nodes – between which the association is established – and their sub-trees. If \( r \) is an association, then

\[
\text{a-grade}(r, \text{imlh}(r)) = \sum_{g \in \text{GP}(r)} w(r, g) H_r(r, g)(\text{imlh}(r)).
\] (3.6.2)

Let

\[
\text{grade}(s, \text{imlh}(s))
\]
\[
= \begin{cases} 
  f\text{-grade}(s, imlh(s)) & \text{if } s \text{ is a functional leaf} \\
  a\text{-grade}(s, imlh(s)) & \text{if } s \text{ is an association} \\
  f\text{-grade}(s, imlh(s)) & \\
  \left( \prod_{t \in (F(s) \cup A(s))} \text{grade}(t, imlh(t)) \right) & \text{otherwise}
\end{cases}
\] (3.6.3)

For any functionality \( f \), the matching grade is defined as

\[
\text{grade}(f) = \max_{imlh \in IMLH} \text{grade}(f, imlh). \tag{3.6.4}
\]

The weights are used to emphasize the geometric properties that are more likely to characterize a specific class. We assume that this likelihood is higher for geometric properties whose histograms have peaks than for those whose value is constant. In order to determine the best geometric properties, we compute the weights as standard deviations [28] of the histograms' values. Note that the standard deviations of the histograms allow the weight mechanism to eliminate unnecessary associations in cliques.

Note also that histograms can be bi-polar, multi-polar, or peak functions. The histogram mechanism covers these cases; however, it is designed and tested for the general case in which the histogram is a non-constant and multi-polar function.

The computation of grades consists of multiplications and weighted summations. We use multiplications on grades of sub-functionalities while we use weighted sums when we want to benefit from the evaluations of the geometric properties. The multiplications are motivated by the existence of functional parts that cooperate toward a higher functionality, which is crucial for classification: you cannot have a chair without a ground-support, for example. However, we expect that when a certain functionality is to be classified, several geometric properties together share a related behaviour, which is expressed in terms of grades.
Our mathematic model assumes that the geometric properties are unrelated. Although this model is not precise, we feel it is sufficient for accurate classification results. Some of the interrelationships among the geometric properties are modeled by weights; however, this issue is outside the scope of this work.

Assume we have a fixed number of primitive parts. Let \( mlh \) be a multi-level hierarchy and \( imlh \) be an implementation of \( mlh \) over this set of primitive parts. Define a partial implementation of \( imlh \) to be a state in which a subset of the leaves of the functional parts in \( mlh \) is implemented by primitive parts. The primitive parts implement the functional parts in a way that conforms to the requirements of the \( imlh \).

Furthermore, let the partial implementations of all possible \( imlhs \) of an \( mlh \) over the given set of primitive parts be the nodes of the search DAG. The root of the search DAG is the partial implementation (we call it the empty \( imlh \)) in which no primitive parts are mapped, while the leaves are all the possible implementations of the \( mlh \).

Assume \( pimlh_1 \) and \( pimlh_2 \) are two partial implementations of \( mlh \) over the set of the primitive parts. We say that \( pimlh_1 \) covers \( pimlh_2 \) if all the primitive parts mapped in \( pimlh_2 \) are also mapped in \( pimlh_1 \) in the same way. \( imlh \) covers all its partial implementations of \( mlh \).

Bearing in mind that the partial implementations of \( mlh \) are covered, we define the edges of the search DAG. Assume \( pimlh_1 \) and \( pimlh_2 \) are two partial implementations of \( mlh \) over the set of primitive parts and \( pimlh_1 \) covers \( pimlh_2 \). Moreover, we require that the difference between \( pimlh_1 \) and \( pimlh_2 \) is that the primitive parts of \( pimlh_1 \), which are not mapped
in \( \text{pimlh}_2 \), are mapped to a simple functional part which is a leaf in the \( mlh \) such that no other primitive parts are mapped with it in \( \text{pimlh}_2 \). Then, we define an \textit{oriented edge}
from \( \text{pimlh}_2 \) to \( \text{pimlh}_1 \) and state that \( \text{pimlh}_1 \) covers \( \text{pimlh}_2 \). We will call this graph the \textit{state search graph}.

The classification phase is a search and validation like algorithm over the state search
DAG (see Figure 3.11). The main difficulty in the classification phase is to efficiently select
the best partitions of the input objects’ primitive parts into functional parts. We focus
on the question “What function could this part fulfil?” For example, if we take a chair,
several plates could be mapped to seat, back support, and one of the legs (as part of the
ground support). Developing algorithms for partitioning and testing their performance is of
great interest.

\subsection*{3.6.2 Matching Implementations of Multi-Level Hierarchies to Functionalities}

We define the matching of implementations of multi-level hierarchies to functionalities as a
search in a finite oriented DAG. Assume that the input 3D model consists of \( n \) primitive
parts whereas the \( mlh \) has \( m \) functional leaves. In this case, the search space is a DAG with
\( n^{m+1} \) leaves.

The search is a traversal of the DAG’s states. We evaluate all the matching grades of all
the implementations in the leaves of the search DAG. The leaf with the highest matching
grade is selected as the one that matches the classification.

When there are many implementations – \( n^{m+1} \), for example – verifying them is a time
Figure 3.11: The search state DAG. This search DAG has a depth of three.
consuming task. Therefore, speedup techniques are in great demand.

We used a heuristic search with a branch-and-bound pruning approach. A good description of the branch-and-bound technique can be found in [32]. Branch-and-bound technique are counterparts of backtracking techniques. The two techniques differ in their search criteria. The search criteria enable the branch-and-bound algorithm to prune branches that cannot provide better results than the intermediary results achieved before the algorithm has to choose among different branches. Next, we define our search criteria.

Consider the notations in Section 3.6.1. Specifically, assume \( f \) is a high-level functionality. In addition, let \( \text{pimlh} \) be a partial \( \text{imlh} \), i.e., an \( \text{imlh} \) for which only some of the functional leaves have primitive parts assigned. Then, for any sub-functionality \( s \) of \( f \) define

\[
\text{f-partial}(s, \text{pimlh}(s)) = \begin{cases} 
\text{f-grade}(s, \text{pimlh}(s)) & \text{if all the functional leaves in } \text{pimlh}(s) \text{ have mapped primitive parts} \\
1 & \text{otherwise}
\end{cases} \tag{3.6.5}
\]

Following 3.6.1, for any association \( r \), let \( \text{pimlh}(r) \) be the sub-part of the \( \text{pimlh} \) that contains the nodes between which the association is established and their sub-trees. If \( r \) is an association, then

\[
\text{a-partial}(r, \text{pimlh}(r)) = \begin{cases} 
\text{a-grade}(r, \text{pimlh}(r)) & \text{if all the functional leaves in } \text{pimlh}(r) \text{ have primitive parts mapped} \\
1 & \text{otherwise}
\end{cases} \tag{3.6.6}
\]

Let

\[
\text{partial-grade}(s, \text{pimlh}(s))
\]
For the search, we use partial matching grades; that is, at each node of the search DAG, we compute the partial grade of the partial implementation available at the current node.

The partial matching grade can only decrease when we evaluate it at subsequent lower level nodes. Moreover, from (3.6.5), (3.6.6), and (3.6.7), it follows that when the search reaches a leaf, the partial grade equals the matching grade.

Following [23], the algorithm searches the nodes of the search graph starting from the “empty" state. The algorithm searches for the implementation of the multi-level hierarchy that has the highest matching grade, which represents the classification result. However, rather than identifying primitive parts, we compare the signatures of functional parts via the geometrical properties of the functional part candidates. When the primitive parts are mapped to functional parts, the matching grades of the functional parts can be computed. For example, we define the volume of a four-legged ground support as being the volume of the four legs sticks together (each leg being a stick, of course).
Chapter 4

Experimental Results

We tested our scheme on a database that includes synthetic models of 200 forks, 216 spoons, 200 stools, and 200 spectacles. We also tested our scheme on a database of real range images comprising 100 forks, 100 spoons, 97 chairs, 100 spectacles, 118 airplane models, 30 cupboards, and 15 tables. Partial sets of the chairs, forks and spoons, airplanes, cupboards, and spectacles are shown in Figures 4.9, 4.10, 4.11, 4.12, and 4.13, respectively. In addition, we used 12 compound objects representing 6 dining rooms and 6 bedrooms (see Figure 4.15). Range images were captured using a Cyberware range scanner (http://www.cyberware.com).

The example of classification is shown in Figure 4.1.

We differentiate between experiments on synthetic data and experiments on real range data. We performed four types of experiments. In the first experiment we checked the performance of the classification. In the second one, we performed a cross validation test and we compute receiver operating characteristics (ROC) for accuracy. We also deal with classification in cluttered scenes. In all the tests, the learning phase was performed on 3D models or range images that contain a single object. In classification performance checking,
Figure 4.1: Example of spectacles classification. (a) The image of the object. (b) The primitive parts detected after segmentation. (c) Result of classification, the resulting functional parts are shown with different colors.

cross validation, and ROC and accuracy tests, we used the proposed scheme to classify objects from 3D models and images that contain only one object. In the cluttered scene tests, partial views of several objects were used.

4.1 Classification Performance

In this classification experiment, three groups of objects were used: a training group and two test sets. The training group was randomly generated from the entire database. The graph in Figure 4.2 shows the test set average grades as a function of the size of the training set. The black curve represents the average grades of the classified spoons. The blue curve shows the ratio between the test sets’ average grade and the maximal grades in the spoons testing sets, in percentages. The learning sets in this experiment consisted of real scanned objects. The test sets of spoons and spectacles is constant per experiment and comprised all the scanned objects, which is the entire database.

Figure 4.2 shows that small training sets suffice for reliable classification. At low abscissa
values, which correspond to small sets of spoons, our scheme produces significantly higher grades for spoons than for spectacles. Furthermore, the average grades of the spectacles grades is less than 0.1, while the black curve, which is the average grades of the spoons, present values that are higher than 0.1.

In an additional experiment, we used a training set of ten chairs where the ground support was implemented using a four-legged structure (see Figure 4.3). We then built a test set that consisted of all the chairs in the database that have ground support, including those built from one and three legs. We correctly classified all the chairs with ground support built
Figure 4.3: Testing classification accuracy. (a) represents a subset of the training set images. (b) illustrates three-legged chairs used in the experiment. All of them were correctly classified. (c) and (d) illustrate one-legged chairs that were used in the experiment. While the chairs in (c) were correctly classified, the scheme could not cope with the object in (d), due to its unusual back-support implemented by a blob.

from three legs. The test set included four chairs with one leg as ground support, and we correctly classified three of them. The chair that was not classified correctly had an unusual back-support implemented by a blob, (see Figure 4.3 (d)). The classifier was set to decide that the input is a chair if the normalized matching grade is over 0.5.
4.2 Cross-Validation

We employ the whole data base in cross-validation experiments. The learning (training) group represented 80% of the object class set whereas the classification (or testing) group consisted of the rest of the data base. The classification involves computing grades: the classified objects are evaluated as spectacles, forks, spoons, mugs, stools, tables, chairs, or airplanes. The grades are averaged and presented as textured bars in a graph, shown in Figure 4.4. The graph consists of eight groups of eight object class grades. For example, the first group relates to spectacles that were classified as spectacles, forks, mugs, spoons, mugs, stools, tables, chairs, or airplanes in this order. In Figure 4.4, the spectacles, the forks, the spoons, the stools, the tables, the chairs, and a subset of the airplanes were range images.

4.3 Receiver Operating Characteristic for Accuracy

We performed experiments on synthetic 3D models as well as on range images. In this section, the first set of experiments employed synthetic 3D models of twenty mugs and twenty spoons (see Figure 4.5). The last set of experiments were performed on the real range images of the entire database (see Figure 4.6).

A receiver operator characteristic curve (ROC) [15] is a graph that allows a user to set one or more thresholds for decision parameters conveniently in order to achieve a certain accuracy. A ROC is related to a classifier that compares a grade of matching to a certain threshold. Specifically, consider a classifier for the class $X$ that for each analyzed object $x$ computes a matching grade and compares it versus a threshold. If the matching grade
Figure 4.4: Cross-validation on the whole data base, which includes spectacles, forks, spoons, mugs, stools, tables, chairs, and airplanes.
is higher than the threshold then it decides that $x \in X$, otherwise it decides that $x \notin X$.

Intrinsically, four sets of objects are formed. These sets are called: true positive, true negative, false positive, and false negative. The true positive set consists of valid instances $x \in X$ that were correctly classified as $x \in X$. True negative objects are ones that are not in the class $x \notin X$ and the classifier correctly classified them $x \notin X$. False positive are objects that are not part of the class $x \notin X$, however, they were detected as part of it $x \in X$. Finally, false negatives are objects that are part of the class $x \in X$ and they are erroneously detected as not being part of the class $x \notin X$. The ROC is defined as the report between the sizes of true positive and false positive sets when the threshold passes from its minimal to its maximum allowed value.

Usually, the parameter tested versus a threshold is bounded by 0 and 1. In this case, ROCs are curves that are begin from the point $(0,0)$ and finishes in $(1,1)$, they being located over the diagonal defined by these 2D points. The quality of such a classifier is the amount of surface they occupy over the diagonal.

We show several measurements of classification in terms of receiver operating characteristic curves (ROC) [15] as well as accuracy tests. In all the graph images, on the abscissas, we show the classification grade thresholds.

4.3.1 ROC: Accuracy on Synthetic Data

In the first set of experiments, we considered a classifier that selects the maximum grades of components and checks if this maximum is achieved for stools and if it is higher than a threshold. We considered a set of 3D synthetic models of mugs and stools. In Figure 4.5...
Figure 4.5: Experiments on synthetic data. Synthetic 3D models of mugs and stools are employed. In this experiment, we considered a classifier that selects the maximum grades for the different components. We present the classification grade threshold on the abscissa.

(a) The classifier’s accuracy of mugs versus mugs and stools. (b) The mug classifier’s accuracy versus that of mugs and stools. (c) Overall accuracy.

We also built classifiers that work on components of the vector matching grades (the classifiers do not make their decisions selecting the maximum on components but by comparing grades to thresholds). We built the ROCs for mugs and spoons separately. These ROCs show almost ideal classifiers, which pass very close to the right top corner of their boundary squares.
4.3.2 ROC: Accuracy on Real Range Image Data

Consider classifiers that work on components of the vector matching grades (the classifiers do not make their decisions by selecting the maximum on components but by comparing grades to thresholds). In Figure 4.6 (a), we show the ROC superimposed curves of stools, forks, and spoons.

Consider a classifier that selects the maximum grades on components, then checks whether this maximum is achieved for stools, and if it is higher than a threshold. In Figure 4.6 (b), we show the classifier’s accuracy of stools versus other objects in the database.

We consider that a general as possible classification setup is of high interest whenever the accuracy of classification schemes is being tested. In Figure 4.6 (c), unlike in Figure 4.6 (a) and (b), we show the accuracy of the classifier when targeting all five tested classes and the whole database.

4.4 Cluttered Scenes

We performed experiments on synthetic 3D models as well as on range images. In this section, we show experiments on synthetic 3D models and real range images separately. The first set of experiments is performed on chairs and forks (see Figure 4.7). The second set is performed on the real range images of the entire database (see Figure 4.8).

4.4.1 Experiments on Synthetic Data

We built 33 synthetic scenes from 3D models of chairs, forks, and spoons. In figure 4.7, we show the classification results for the chair in a cluttered environment consisting of the
Figure 4.6: Experiments on real range image data. We show the classification grade threshold on the abscissa. (a) A receiver operating characteristic on the whole data base for stools (the uppermost curve - dashed) forks (the middle curve - normal), and spectacles (the lowest curve - dot and dashed). (b) The classifier’s accuracy of stools versus other objects in the database. (c) Overall accuracy.

chair and a synthetically enlarged fork. The synthetically enlarged fork is not a usual object. Scaling drastically diminishes the grade for a fork as a fork. Moreover, all the component grades for the fork as an object belonging to other classes are low. The grade received by the chair as a chair is much higher. Therefore, the classifier makes a valid conclusion about the occurrence of a chair.

4.4.2 Experiments on Real Range Image Data

In this section, the learning phase consisted of images that included only one chair. In the classification phase we tested our scheme using two types of 3D images. The first type consisted of a chair and a table in a room while the second type consisted of chairs and synthetically enlarged spoons (see Figure 4.8 for examples of first type of images). We tested our scheme on six images of the first type and thirty images of the second type. In
five images of the first type and in all the images of the second type, our system correctly classified a valid chair. One image of the first type has heavy cluttering and significant self-occluding regions. This fact caused our scheme to misclassify the target.

4.5 Classifying Compound Objects

The aim of this experiment is to demonstrate the ability to generalize to complex objects. We try to recognize dining rooms and bedrooms. Room is a class with many subclasses, or in other words, it has many specializations.

We used six rooms that contain a chair and a table and we characterize them as dining rooms (see Figure 4.15 (a)). In addition, we used six rooms with a chair and a bed. We refer to these six rooms as bedrooms (see Figure 4.15 (b)). Note that the use of the chair in a room environment implies a multi-level hierarchy with height four (see Figure 4.14). Here, the bed is an indication of a bedroom. In one dining room, the chair was not recognized due to heavy
Figure 4.8: Two cluttered scenes of a chair and a table in a room. (a) and (b) represent two examples of the first type of image. Figure (c) and (d) represent the results of segmentation of the scenes in (a) and (b), respectively. The primitive parts sticks and plates are shown as they are detected and modelled in the segmentation phase. Figure (e) and (f) represent the results of classifying the scenes in (a) and (b), respectively. (e) and (f) show the resulting functional parts – the back support, seat, and the ground support – with different textures.
Figure 4.9: Images of some chairs used in the experiments.
cluttering, and so we could not classify it. However, whenever we could classify the objects in
the room, we were able to clearly differentiate between dining rooms and bedrooms. Figure
4.16 shows an image of a bedroom with classifications results. The classification of the chair
and of the bed is shown Figure 4.16.

When working in compound scenes one can use the advantages of the reusable learning
provided by function-based reasoning. For example, when learning dining rooms and bed-
rooms, the common denominator is learning chairs. This shared learning is illustrated in
Figure 4.17.
Figure 4.11: Images of several airplanes
Figure 4.12: Images of several cupboards
Figure 4.13: Images of some spectacles used in the experiments.
Figure 4.14: Illustration of an $mlh$(Bedroom) representation with height four.

### 4.6 Implementation Details

Almost any computer vision system assumes a number of parameters that have to be tuned for specific applications. We acknowledge that the segmentation part of our scheme is dependent on the input data. However, we report that the functional part is generic enough to enable classification without human intervention, provided the learning stage is supervised.

Let $B_{i,k,(t)}$ be the $i$th B-spline blending function of degree $k$ defined over knot sequence $\tau$ [19]. Now, consider the B-spline function,

$$ f(u) = \sum_{i=0}^{n} p_i B_{i,k,(t)}(u), $$

with $n+1$ scalar coefficients $p_i$, B-spline basis functions $B_{i,k,(t)}(u)$, degree $k$ and knot sequence $\tau$, respectively. We use uniform knot sequences and employ cubic B-splines to implement
Figure 4.15: Examples of two bedrooms and two dining rooms we used in experiments.
Figure 4.16: Classification of a bedroom. By definition, a bedroom includes at least a bed. In this example, we show the classification of bedrooms versus dining rooms. (a) represents a digital photo of a room with a chair and a bed. The magenta color represents unselected parts during the classification process. The image in (b) represents the classification of the room as a bedroom, after the chair and the bed are selected (and colored) for analysis.

Figure 4.17: Illustration of reusable function based learning.
the histograms of geometric properties employed in the operational multi-level hierarchies, (see Section 3.5 and [19]).

We use uniform knot sequences and employ cubic B-splines in this work. These B-spline functions allow the computation of matching grades between the multi-level hierarchy implementations and the operational ones.

Unlike the authors of [12, 35], we implemented histograms (as RBF functions) via B-spline functions and not Gaussian mix because the former exhibited better time performance (see [19]). In addition, the fitting of B-spline functions is generic for each geometric property. In contrast, fitting Gaussians implies parameter tuning for their average and the divergence. Nevertheless, B-spline functions provide reliable accuracy (see [19]).
Chapter 5

Conclusions

In this work, we have presented a novel function-based scheme for classification of 3D objects. The input objects are full 3D descriptions of objects. The proposed scheme employs an object functional structure and consists of a multi-level hierarchy of functional parts. The multi-level approach offers a higher degree of freedom for real object modelling than is possible in classical systems. The multi-level hierarchy implementation represents a supervised learning phase.

Our approach was tested on a database of about one thousand different 3D objects and employed several algorithms for searching and pruning. To the best of our knowledge, no other classification (or recognition) scheme has been tested on hundreds of real objects captured in range images. Automatic segmentation usually suffers from over-segmentation. This phenomenon does not influence the accuracy of the proposed solution, however, it could produce an increase of the time complexity of the searching phase. The graphs show the success of our scheme. They also provide an insight into the dimensions of the learning sets that are required to reach a certain degree of classification accuracy. Moreover, we have also
demonstrated how reusable function based learning can benefit our function based reasoning scheme. To the best of our knowledge no such reusing was used nor described in existing systems.

A possible direction for future research is to enlarge the data base of the test objects. Specifically, in future experiments, additional categories of range image objects could be introduced. In addition, the usage of more accurate approximation models for primitive parts and use more elaborate models to describe these parts should be studied. However, it should be noted that the use of relatively coarse parts had no negative influence on recognition of difficult categories.

In our opinion, an interesting direction of future work is introduction in the scheme “virtual” primitive parts. For example, holes in handles are crucial for functionality in scissors or shears (see Figure 5.1). This holes could be described with non existing in image “virtual” primitive part.

Another interesting direction could be the fusion of data from different modalities. Amselem et al. [1] provides example of enhancement of our scheme with sound modality. Since the color information could provide knowledge about used materials etc. it could be used for one more enhancement of the proposed system.

The proposed solution is clearly parallelizable; concurrent or parallel variants of our scheme as well as implementations of our classification algorithm on dedicated hardware could greatly speed up the classification process.

A very interesting direction for future research may be introducing new geometrical
approaches that are intrinsic to shape such as computing curvatures and other differential characteristics.

As a final proposal for future research we should mention the building of a robot equipped with recognition system based on the proposed scheme. This robot could be used for interactive verification of obtained functional based classification. It also could be used to check the function based environment understanding approach proposed in our research.
Acknowledgments

The author thank Octavian and Diana Soldea, Nela Gurevich, Emil Dan Kohn, Lev Ferdinskoif, Ohad Serfaty, Vitali Greenberg, Artyom Sharov, Yanai Shpinner, Fady Massarwi, Ebraheem Kassem, Adir Abraham, Ronen Keidar, and Yekutiel Pekar for their support. A special thanks to my wife Olga Pechuk for love, patience and support.
Appendix A

Geometric Properties Used in Classification
<table>
<thead>
<tr>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>CircularRegularityByAngle</td>
</tr>
<tr>
<td>CircularRegularityByRadius</td>
</tr>
<tr>
<td>FPartConnectivity</td>
</tr>
<tr>
<td>LinearRegularity</td>
</tr>
<tr>
<td>Stability</td>
</tr>
<tr>
<td>LinearSymmetry</td>
</tr>
<tr>
<td>CircularSymmetry</td>
</tr>
<tr>
<td>RegularSurface</td>
</tr>
<tr>
<td>RegularVolume</td>
</tr>
<tr>
<td>BBoxSurfaceArea</td>
</tr>
<tr>
<td>PPartsMajorAxesStandardDeviation</td>
</tr>
<tr>
<td>InertiaMomentsIxx</td>
</tr>
<tr>
<td>InertiaMomentsIxy</td>
</tr>
<tr>
<td>InertiaMomentsIxz</td>
</tr>
<tr>
<td>InertiaMomentsIyy</td>
</tr>
<tr>
<td>InertiaMomentsIyz</td>
</tr>
<tr>
<td>InertiaMomentsIzz</td>
</tr>
<tr>
<td>InertiaEigenValue1</td>
</tr>
<tr>
<td>InertiaEigenValue2</td>
</tr>
<tr>
<td>InertiaEigenValue3</td>
</tr>
<tr>
<td>BBoxVolume</td>
</tr>
<tr>
<td>RelativeVolumeBBoxVolume</td>
</tr>
<tr>
<td>RelativeSurfaceBBoxSurface</td>
</tr>
<tr>
<td>COMDistanceFromBBoxCenter</td>
</tr>
<tr>
<td>CenterOfMassWithinBBox</td>
</tr>
<tr>
<td>FPartOrientation</td>
</tr>
</tbody>
</table>

Table A.1: Geometric properties for primitive and functional parts.
<table>
<thead>
<tr>
<th>CrucialConnectivity</th>
<th>CrucialStability</th>
</tr>
</thead>
<tbody>
<tr>
<td>RelativeFPartCenterOfMass</td>
<td>RelativeFPartVolume</td>
</tr>
<tr>
<td>RelativeSurface</td>
<td>RelativeFPartBBoxVolume</td>
</tr>
<tr>
<td>RelativeFPartBBoxSurface</td>
<td>MassCentersRelativeDirection</td>
</tr>
<tr>
<td>MassCentersRelativeDistance</td>
<td>NumberOfPPartConnections</td>
</tr>
<tr>
<td>TypeOfConnection</td>
<td>ConnectionGeometricDescription</td>
</tr>
<tr>
<td>BBoxConnection</td>
<td></td>
</tr>
</tbody>
</table>

Table A.2: Geometric properties for connections.
Bibliography


[38] www.cs.technion.ac.il/~mpechuk/ProjectOCLS/index.html


[53] A. Turing. Computing Machinery and Intelligence, Mind 59, pp. 433460, 1950


