



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

Computer Vision and Image Understanding xxx (2007) xxx–xxx

Computer Vision  
and Image  
Understanding[www.elsevier.com/locate/cviu](http://www.elsevier.com/locate/cviu)

# Learning function-based object classification from 3D imagery

Michael Pechuk, Octavian Soldea \*, Ehud Rivlin

*Department of Computer Science, The Technion, Israel Institute of Technology, Haifa, Israel*

Received 5 December 2006; accepted 15 June 2007

## Abstract

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 mechanisms. 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 the class that provides 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 1000 different 3D objects. The proposed scheme achieved high classification accuracy while using small training sets.

© 2007 Elsevier Inc. All rights reserved.

*Keywords:* Function-based reasoning; Object classification; 3D range data; Supervised learning

## 1. Introduction

The problem of object classification from sensory data is defined, in literature, as the association of visual input with 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). We propose a new 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. The learning phase is notably accelerated as a result.

We tested the proposed scheme on a database of about 1000 different 3D objects. No other classification (or recognition) scheme has yet been tested on hundreds of real objects captured in range images.

Our paper is organized as follows: in Section 2, we present an overview of the literature on function-based

\* Corresponding author.

*E-mail address:* [octavian@cs.technion.ac.il](mailto:octavian@cs.technion.ac.il) (O. Soldea).

recognition from 3D data. Next, in Section 3, we describe the proposed method. We present the details of segmentation (Section 3.1), the multi-level hierarchy functional structure (Section 3.2), the supervised learning method (Section 3.5), and the proposed classification method (Section 3.6). In Section 4, we present the experiments we performed and analyze the accuracy of our results. We present our conclusions in Section 5.

## 2. Related work

The computer vision community adopted fundamental ideas from psychology, artificial intelligence, and linguistics in order to achieve automatic understanding of images. Following [31], the authors tried to provide implementations for frames of knowledge.

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 [12,13,19,21,33,35,50]. Later on, parameterized geometric modeling was introduced [23,46,48]. 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. Examples for such an approach can be found in [2,6,7,27,36].

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, 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 that 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. Such approaches, which are fundamentally different from the previous ones, were introduced in [11,16,51].

An impressive number of results in the function-based classification field were demonstrated with the GRUFF and OMLET systems. The authors in [41,43], 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.

The authors in [17] describe a function-based recognition system for dealing with objects whose function depends on parts joined by articulated connections. The first attempts at categorizing from incomplete 3D shape object descriptions were made in [42], where the authors describe an application of the GRUFF concepts of function-based reasoning to an OPUS model (object plus unseen space). More recently, the authors in [44] built a testing framework for the GRUFF system using stereovision images. However, recognition accuracy is not detailed. Note that GRUFF was developed for the learning of membership functions from 3D objects [52]. 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 [11] and [51]. Following these concepts, several systems for object classification were built (see [1,11,41,43]). However, little experimental work has been done to test these concepts. Only preliminary attempts were made toward functional classification from raw images of objects [42] and stereovision-based models [44]. 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 [4,37]. Comparing implemented schemes to human abilities [47] is important as well. Following [47], the authors of [40] 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.

In [28], the authors 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.

The authors in [1] propose a modeling 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 in [39]. The authors in [38] 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).

The authors in [49] 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 in [20], 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. More recently, the authors in [25] propose a generic model in which inferences are drawn from examples. Contributions to the functional reasoning domain, such as physics-based analysis, experimentation of different verification stages, as well as reverse engineering can be found in [5,15,38].

A good overview of function-based classification methods can be found in [3]. The authors feel that common sense reasoning is a particular, specialized, and very high level kind of functional reasoning [9,30,32].

Following the pioneering work of Marvin Minsky [31], the computer vision community tried to further refine and implement his ideas (see [23,33,35]). In this context, functional-based recognition is probably the newest refinement in this field. Our approach involves decomposition of the analyzed objects to primitive parts, a fact that emerges from the way in which human being process the objects in the environment. We begin from a low-level processing stage, in which primitive parts are grouped together (and therefore a parallel like processing is undertaken), to a higher stage, where functional cues are transformed in computable (almost sequential) procedures. We believe that this is the most natural way to model the human understanding process, i.e., targeting functional objectives or goals, we begin from parallel computations of primitives to sequential validations of high-level primitives. To the best of our knowledge and for the first time, we present, a full general scheme of functional reasoning in terms of multi-level hierarchies, which we propose as a convenient tool for symbolic signatures-based reasoning.

Our scheme represents a supervised learning algorithm. To the best of our knowledge, OMLET is the first and sole system that involved elements of learning in the context of function-based reasoning (see [52]). OMLET employed predefined characteristic functions that were fine-tuned during learning. Unlike OMLET, in the learning stage, we are able to create any characteristic functions seems to be appropriate. Our supervision stage means providing segmented and tagged examples.

We performed an as extensive as possible plethora of experiments. Moreover, we intended to and described a complete and an as general as possible classification scheme. We described the way in which we computed classification grades in a clear and concentrated sequence of mathematical, and therefore reproducible, formulae.

### 3. Learning and classifying functionalities

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 [38], 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. Thousands of objects can be mapped to a structure consisting of only a few primitive parts. The immense number of objects in nature is an expression 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 Fig. 1). 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 Fig. 2). During learning, the values of the geometric properties are merged together in signatures implemented by RBF (radial-based functions) [8,29]. 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

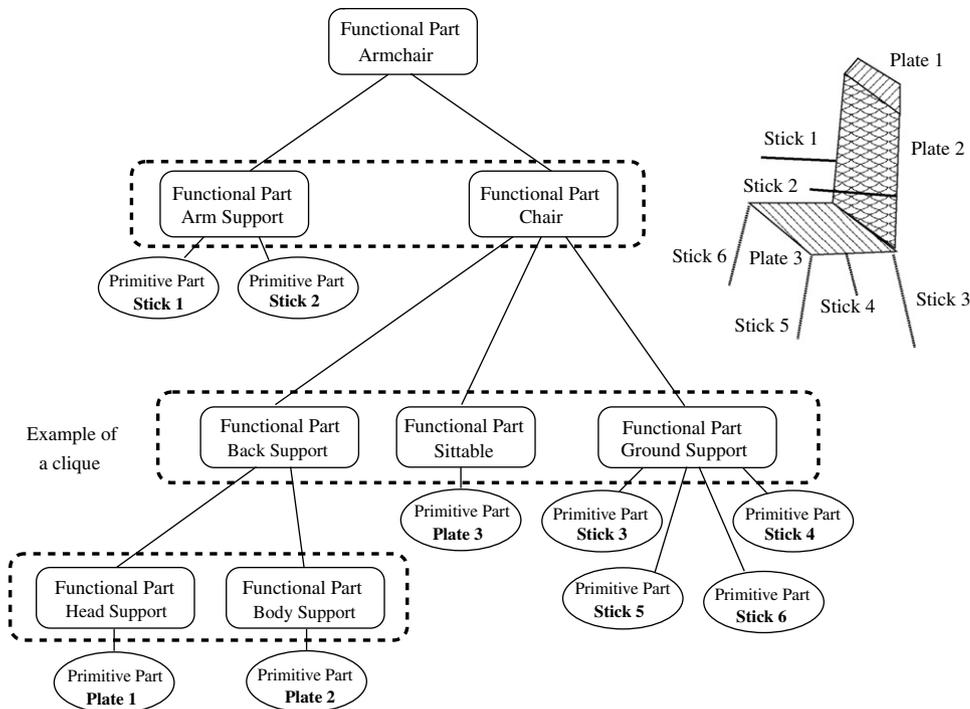


Fig. 1. An implementation of the multi-level hierarchy of functionalities that describe an “Armchair”.

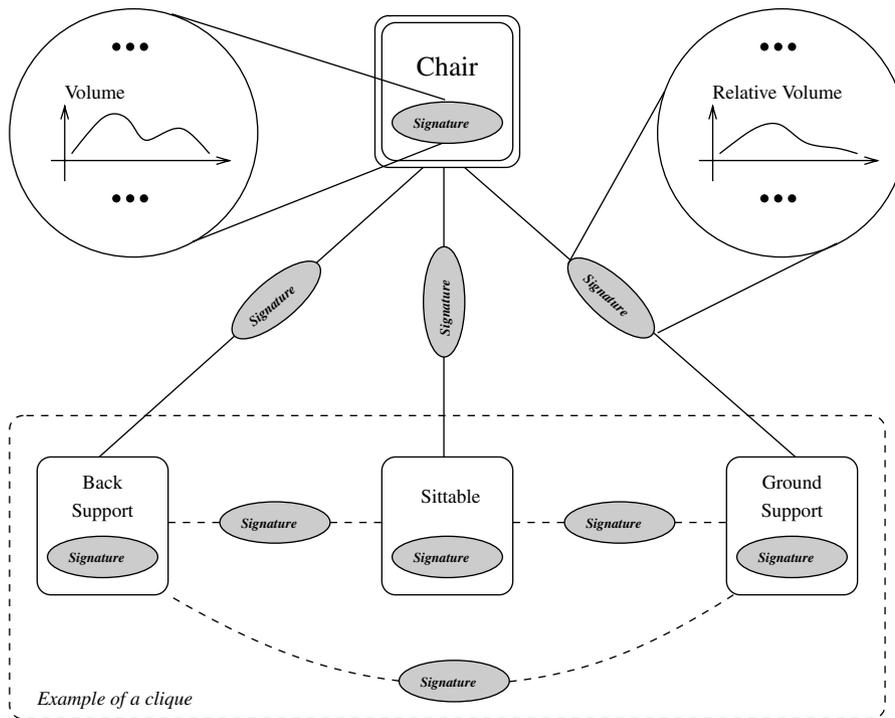


Fig. 2. An operational multi-level hierarchy of a chair denoted  $omlh(\text{Chair})$ . The nodes of  $omlh(\text{Chair})$  are sub-functional parts of the object Chair, while the signatures are sets of likelihood functions built from histograms. These functions provide grades for matching different parts of the entire layered functional structure.

structure). The range images were captured using a Cyberware range scanner (<http://www.cyberware.com>). 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 [22]. This work summarizes four range segmentation algorithms and presents a comparative framework for testing and comparing them. The authors of [22] conclude that the so-called

UE (University of Edinburgh) algorithm, based on Gaussian and mean curvature estimation, provides the best results despite its being the most time-consuming scheme. We provide in Algorithm 1 a sketch of the used algorithm.

---

**Algorithm 1.** An adapted version of the UE algorithm

---

**Input:**

A raw range image

**Output:**

A set of primitive parts

**Adapted\_UE\_Segmentation\_Algorithm**

- 1: **for all** point  $P$  on the grid **do**
  - 2:   Consider a small neighborhood  $Patch(P)$  of grid points around  $P$
  - 3:   Approximate  $Patch(P)$  with a plane  $Plane(P)$
  - 4:   Compute the normal  $Normal(P)$  of  $Plane(P)$
  - 5: **end for**
  - 6: Smooth the data preserving the large discontinuities in the normal angles as well as in the depth information
  - 7: **for all** point  $P$  on the grid
  - 8:   Compute the curvatures  $K(P)$  and  $H(P)$
  - 9: **end for**
  - 10: Collect points  $P$  in small segments analyzing curvature signs.
  - 11: Perform region growing on segments, eventually unifying them, employing erosion–dilation morphologic techniques
  - 12: Approximate the segments as sticks, plates, and blobs using up to second order moments
- 

The learning and the classification phases receive as input primitive parts as detected by the segmentation algorithm, which is a variation on the UE (University of Edinburgh) algorithm [22]. This algorithm was 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 a major issue, the UE algorithm provided reliable results in the framework of our experiments. We employed it for constructing representation models for the primitive parts. Segmentation results for a range image of a plastic airplane model are shown in Fig. 3.

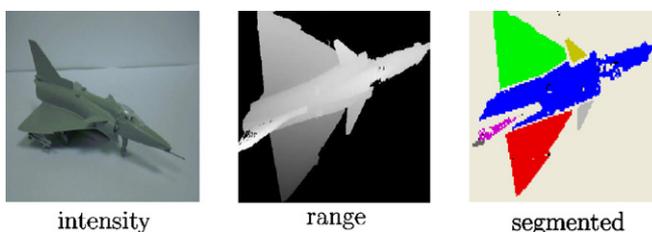


Fig. 3. Segmentation of a valid airplane results in one deformed stick and five plates.

### 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 Fig. 1). 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 [38]. This approach is known in literature as recognition/classification by functional parts.

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 of 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 Fig. 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.

The connections between the primitive parts are obtained during the segmentation stages. Note that segmentation can provide cues about the connectivity and the occlusion of 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 Fig. 4, Sticks 1–4 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 Fig. 4 for an example of an  $mlh(\text{Chair})$ ]. 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.

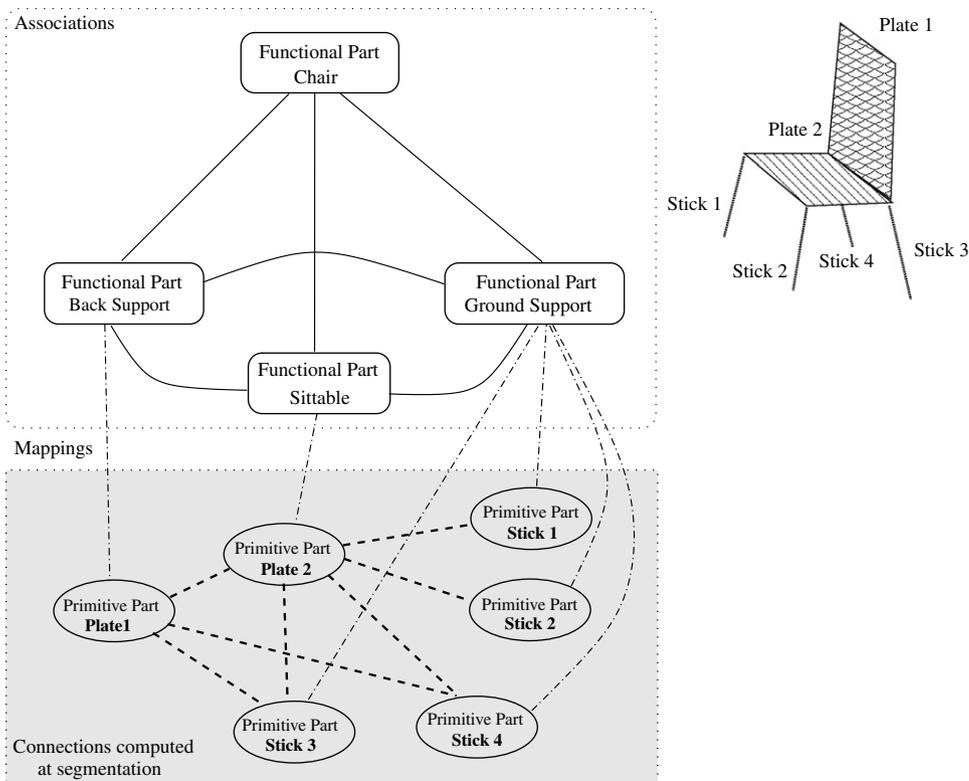


Fig. 4. Associations, connections, and mappings shown on an implementation of a multi-level hierarchy of a chair. These three types of relationships are represented by bold, dashed, and dashed-and-dotted lines respectively.

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 nodes of the functional and primitive parts, and the relationships among the parts (see Figs. 1 and 4). We denote the implementation of the multi-level hierarchy of  $f$  by  $imlh(f)$ .

If  $s$  is a functional node in the  $imlh$  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 Fig. 5). If  $s$  is an association or a connection, then let  $imlh(s)$  be the edge-like sub-

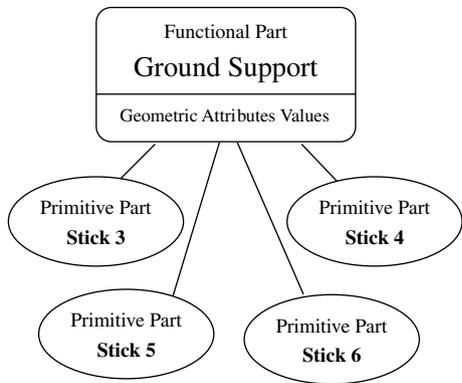


Fig. 5.  $imlh(\text{Ground Support})$  is the implementation of the sub-functional part of the armchair  $imlh$  described in Fig. 1.  $imlh(\text{Ground Support})$  is the  $imlh$  of a part of the entire  $imlh(\text{Armchair})$ .

structure of  $imlh$  that consists of all the geometric property values that are constituents of this relationship; see Fig. 6. Of course,  $imlh$  and  $imlh(f)$  are equal.

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. The pairs of functional parts (in the

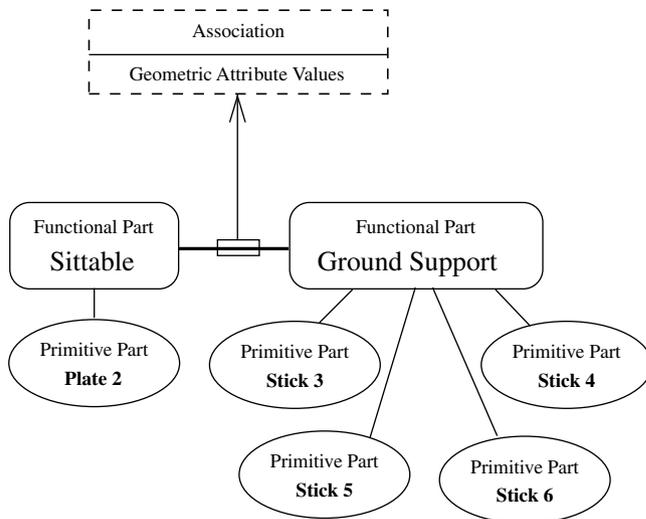


Fig. 6. An example of  $imlh(A(\text{Sittable}, \text{Ground Support}))$ . Here,  $imlh(A(\text{Sittable}, \text{Ground Support}))$  is the  $imlh$  of a part of the entire  $imlh(\text{Armchair})$  and  $A(x,y)$  refers to the associations between the functional parts  $x$  and  $y$ .

clique) are characterized by a relationship expressed in terms of geometric properties.

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 Figs. 1 and 7. Note that  $s$  can be  $f$  itself.

In Fig. 1, 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 Fig. 1 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  $P(s) = \emptyset$ ,  $F(s) \neq \emptyset$ .

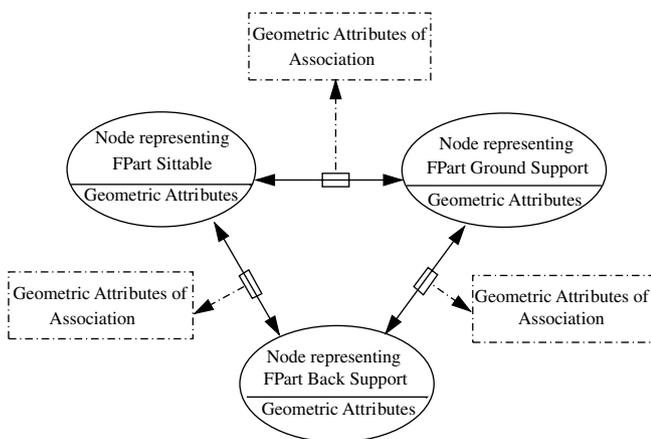


Fig. 7. A clique in the multi-level hierarchy functional structure. The clique corresponds to the group of functional parts Back Support, Sittable, and Ground Support, enclosed in a dashed contour boundary in Fig. 1. These functional parts of the functional part chair are siblings, where the functional 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.

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 ( $C^0$  continuity). If  $x$  is an association,  $GP(x)$  includes, among other properties, ratio of volumes and context-based stability.

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 [34].

### 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

$$GP_{PF} = \{(x, g) | x \in F \cup P, g \in GP(x)\}$$

and

$$GP_{CA} = \bigcup_{f \in F} \{(y, g) | y \in C(f) \cup A(f), g \in GP(y)\}.$$

Then, the multi-level hierarchy of a functionality  $f$  induces a function  $\mathcal{H}_f : GP_{PF} \cup GP_{CA} \rightarrow H$ , where  $H = \{h | h: R \rightarrow [0..1]\}$  is the set of all (normalized) histograms that can be implemented as B-spline functions (see Sections 3.5 and 3.7). These histograms are normalized to 1. Therefore, they are likelihood functions. Evaluating  $\mathcal{H}_f$  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 functional part, the set of histograms of its constituents, functional (sub)-parts, and associations represent 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 Fig. 2. 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 learned 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 Fig. 8.

### 3.5. Learning functionalities

The left-hand side of Fig. 9 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 predefined

geometrical properties. Furthermore, these properties are subject to RBF-like (radial-based function) learning [8,29]. The result is an operational multi-level hierarchy for each learned object.

### 3.6. Classification

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 3.7). The continuous domain

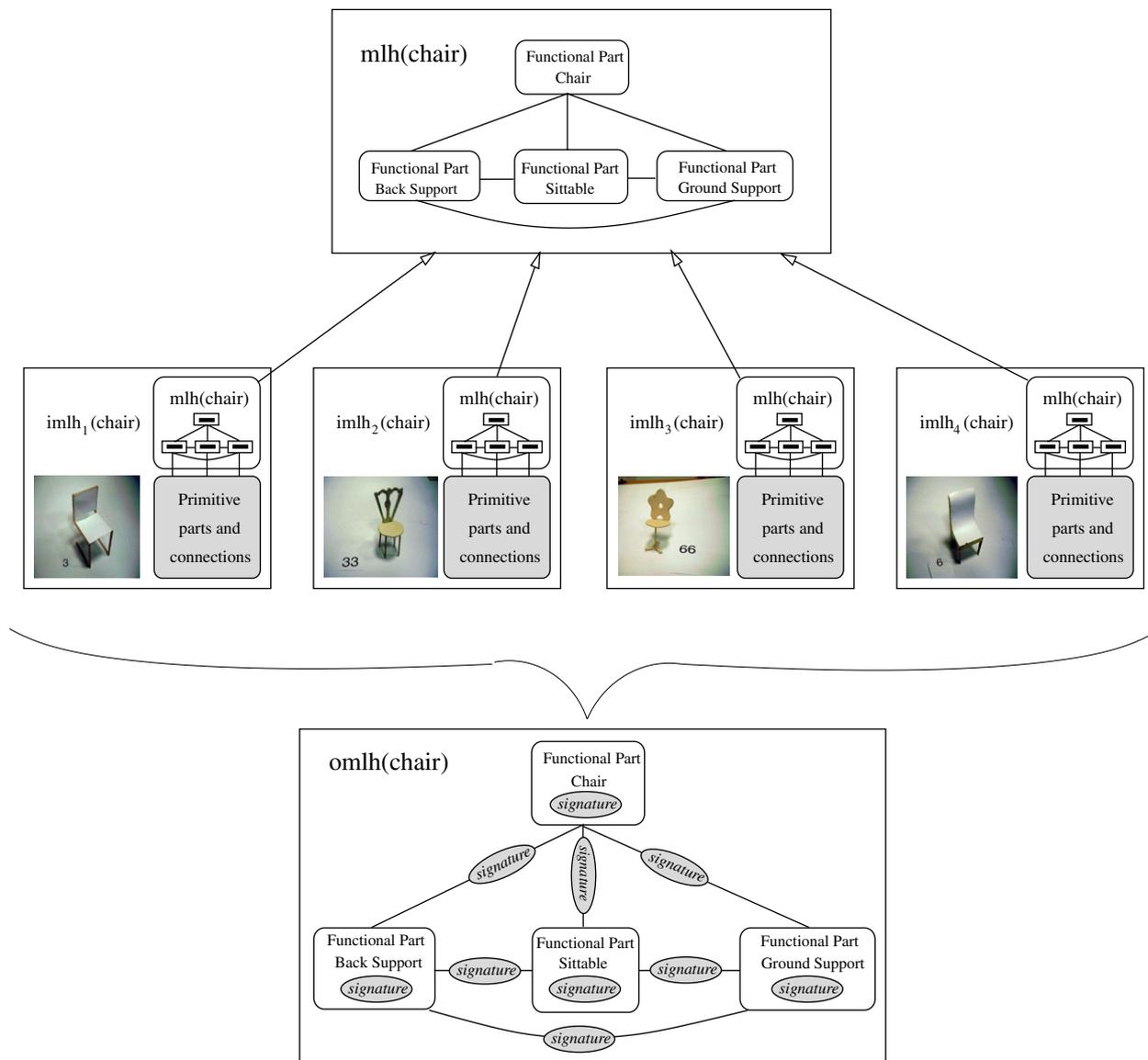


Fig. 8. Construction of several *imlhs*, an *omlh*, and their use in the case of a chair. Each image in the testing set is segmented and an *imlh* is constructed for it. These *imlhs* are input to the learning stage, in which we compute an *omlh*. At classification, the analyzed image is segmented to primitive parts and the grade of matching to any learned class is evaluated by means of the *omlh*.

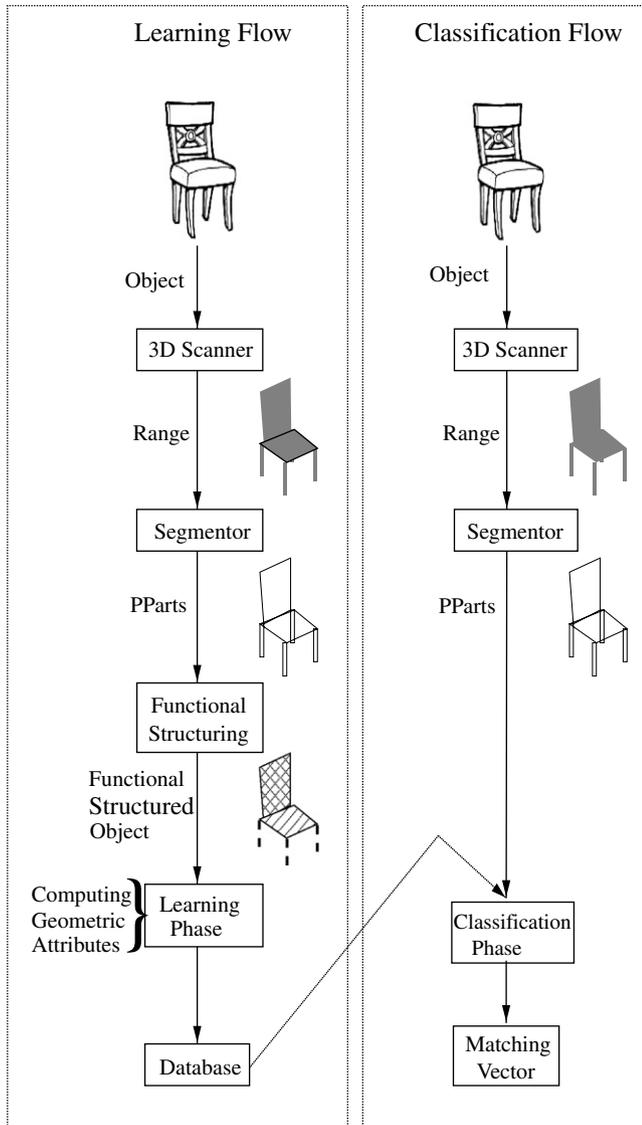


Fig. 9. Learning and classification flows.

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 Fig. 1, 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.

Fig. 9 (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 learned 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 subsections describe the matching grade computation process as well as the mathematical 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  $IMLH$  be the set of all implementations of instances of  $mlh$  built with primitive parts from  $Prim$ .

Recall that we use the notation  $imlh(s)$  for the sub-tree-like layered structure which has  $s$  as a root and is part of  $imlh$ , where  $s$  is a functional node in the  $mlh$  of  $f$  (see Fig. 5). Moreover, we also use the notation  $imlh(s)$  for the (edge-like) substructure of the  $imlh$  that consists of all the geometric property values that are constituents of this relationship, whenever  $s$  is an association or a connection; see Fig. 6.

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 [24] of the histogram function which itself corresponds to  $s$  and  $g$ . Let  $\mathcal{H}_s(s, g)(imlh(s))$  be the value of the geometric property histogram for  $s$  implemented after the implementation  $imlh(s)$ . Then, define

$$f\text{-grade}(s, imlh(s)) = \sum_{g \in GP(s)} w(s, g) \mathcal{H}_s(s, g)(imlh(s)). \quad (1)$$

In this context, for any association  $r$ , let  $imlh(r)$  be the subpart of the  $imlh$  that contains the nodes—between which the association is established—and their sub-trees. If  $r$  is an association, then

$$a\text{-grade}(r, imlh(r)) = \sum_{g \in GP(r)} w(r, g) \mathcal{H}_r(r, g)(imlh(r)). \quad (2)$$

Let

$$grade(s, imlh(s)) = \begin{cases} f-grade(s, imlh(s)) & \text{if } s \text{ is a functional leaf} \\ a-grade(s, imlh(s)) & \text{if } s \text{ is an association} \\ f-grade(s, imlh(s)) \left( \prod_{t \in (F(s) \cup A(s))} grade(t, imlh(t)) \right) & \text{otherwise.} \end{cases} \quad (3)$$

For any functionality  $f$ , the matching grade is defined as

$$grade(f) = \max_{imlh \in MLH} grade(f, imlh). \quad (4)$$

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 [24] 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 mathematical model assumes that the geometric properties are unrelated. Although this model is not precise, we feel it is sufficient for accurate classification. 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 primitive parts of  $pimlh_1$ , which are not mapped in  $pimlh_2$ , be mapped to a simple functional part which is a leaf in the  $mlh$ . The primitive parts are mapped to a simple functional part which is a leaf in the  $mlh$  to which no other parts in  $pimlh_2$  are mapped. Then, we define an *oriented edge* from  $pimlh_2$  to  $pimlh_1$  and state that  $pimlh_1$  covers  $pimlh_2$ . We will call this graph the *state search graph*.

The classification phase is a search and validation like algorithm over the state search DAG (see Fig. 10). The main difficulty in the classification phase is to efficiently select the best partitions (or matchings) of the input objects' primitive parts into (to) functional parts. Moreover, we focus on the question "What function could this part fulfil?" For example, in the case of a chair, several plates could be mapped to seat, back support, and one of the legs (as part of the ground support). In order to answer this question, we present, in Section 3.6.2, the partitioning mechanism (we call this, mechanism matching).

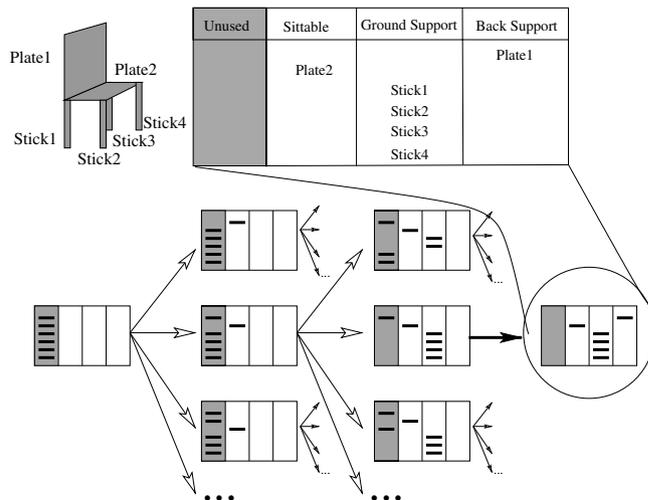


Fig. 10. The search state DAG. This search DAG has a depth of three.

### 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-consuming task. Therefore, speedup techniques are in great demand.

We used a heuristic search with a branch-and-bound pruning approach. Note that branch-and-bound techniques use search criteria and a good description of the this technique can be found in [26]. 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 *pimlh* be a partial *imlh*, i.e., an *imlh* for which only some of the functional leaves have primitive parts assigned. Then, for any sub-functionality  $s$  of  $f$  define

$$f\text{-partial}(s, pimlh(s)) = \begin{cases} f\text{-grade}(s, pimlh(s)) & \text{if all the functional} \\ & \text{leaves in } pimlh(s) \\ & \text{have mapped} \\ & \text{primitive parts} \\ 1 & \text{otherwise.} \end{cases} \quad (5)$$

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

$$a\text{-partial}(r, pimlh(r)) = \begin{cases} a\text{-grade}(r, pimlh(r)) & \text{if all the functional} \\ & \text{leaves in } pimlh(r) \\ & \text{have primitive} \\ & \text{parts mapped} \\ 1 & \text{otherwise.} \end{cases} \quad (6)$$

Let

$$partial\_grade(s, pimlh(s)) = \begin{cases} f\text{-partial}(s, imlh(s)) & \text{if } s \text{ is a functional leaf} \\ a\text{-partial}(s, imlh(s)) & \text{if } s \text{ is an association} \\ f\text{-partial}(s, imlh(s)) & \\ \left( \prod_{t \in (F(s) \cup A(s))} partial\_grade(t, imlh(t)) \right) & \text{otherwise.} \end{cases} \quad (7)$$

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 (5)–(7), it follows that when the search reaches a leaf, the partial grade equals the matching grade.

Following [18], 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 leg-sticks together (each leg being a stick, of course).

### 3.7. 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, the functional part is generic enough to enable classification without human intervention, provided the learning stage is supervised.

In order to avoid parameter tuning and enable generic software, we employed B-spline functions. We use uniform knot sequences and employ cubic B-splines to implement the histograms of geometric properties employed in the operational multi-level hierarchies (see Section 3.5 and [14]). Let  $B_{i,k,\tau}(t)$  be the  $i$ th B-spline blending function of degree  $k$  defined over knot sequence  $\tau$  [14]. Now, consider the B-spline function,

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

with  $n + 1$  scalar coefficients  $p_i$ , B-spline basis functions  $B_{i,k,\tau}(u)$ , degree  $k$  and knot sequence  $\tau$ , respectively. These B-spline functions allow the computation of matching grades between the multi-level hierarchy implementations and the operational ones.

Unlike the authors of [8,29], we implemented histograms (as RBF functions) via B-spline functions and not Gaussian mix because the former exhibited better time performance (see [14]). 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 [14]).

#### 4. Experiments

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, and 15 tables. Partial sets of the chairs, forks and spoons, airplanes, and spectacles are shown in Figs. 11–14, respectively. In addition, we used 12 compound objects representing six dining rooms and six bedrooms (see Fig. 15). Range images were captured using a Cyberware range scanner (<http://www.cyberware.com>). Note that the



Fig. 13. Images of several airplanes.

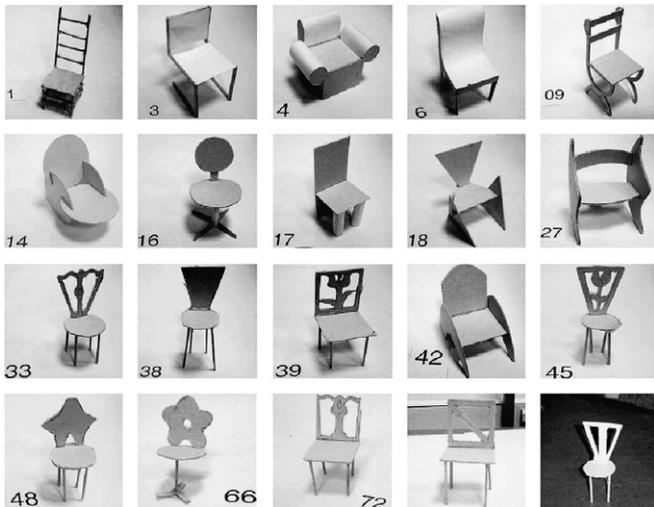


Fig. 11. Images of some chairs used in our experiments.

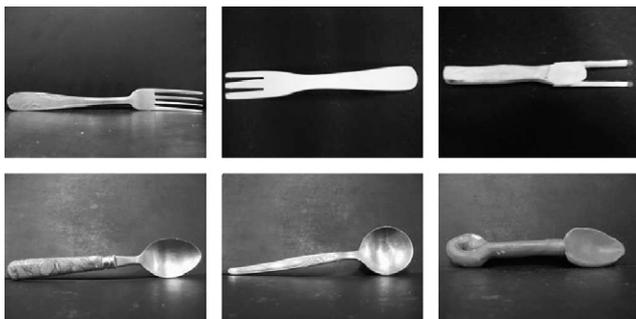


Fig. 12. Images of some forks and spoons used in our experiments.

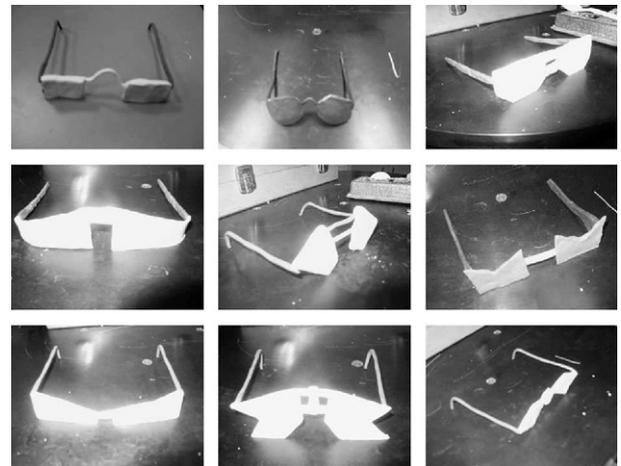


Fig. 14. Images of some spectacles used in our experiments.

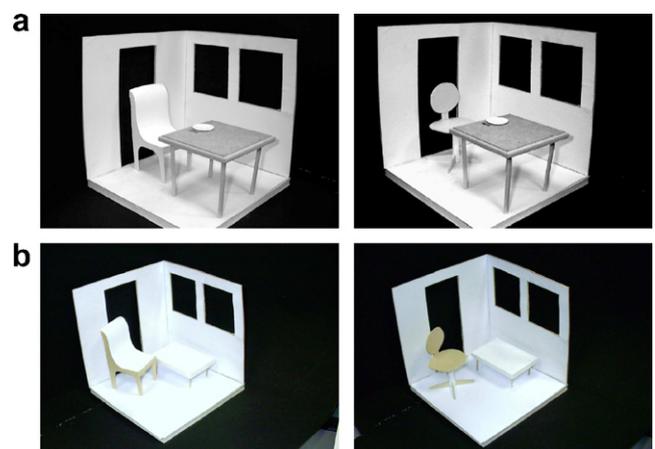


Fig. 15. Examples of two bedrooms and two dining rooms we used in experiments.

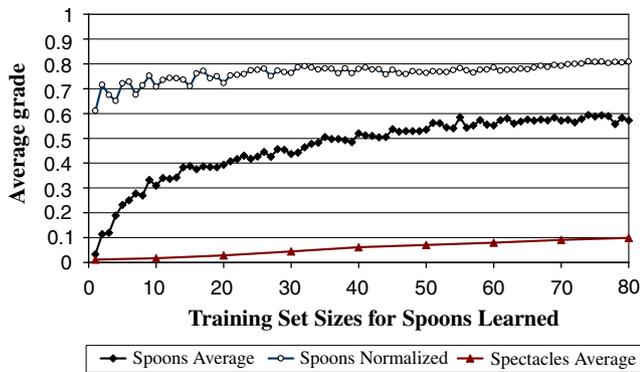


Fig. 16. Learning real spoons and classifying real spoons and spectacles. The average grades of the spoons are all above 0.1 while those of the spectacles are all below that value. This assures very accurate classification.

chairs and the tables were toy-sized models but scaled to characteristic dimensions before any processing stage.

We differentiate between experiments on synthetic data and experiments on real range data. We performed six types of experiments. In the experiments of the first type, we checked the performance of the classification algorithm. In those of the second type, we performed cross-validation tests. Next, we computed receiver operating characteristics (ROC) and accuracy measures. We then dealt with classification in cluttered scenes. Next, we focused on classifying compound objects. In the experiments of the last type, we provide a glimpse on the effects of over-segmentation on the classification results. In all the tests, the learning phase was performed on 3D models or range images that contained a single object. For classification performance, cross-validation, ROC and accuracy, and over-segmentation tests, we used the proposed scheme to classify objects from 3D models and images that contained only one object. In the cluttered scene tests, partial views of several objects were used.

#### 4.1. Classification performance

In the first 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 at consequent increasing sizes. The graph in Fig. 16 shows the average grades of the test sets as a function of the size of the training set. The black curve represents the average grades of the classified spoons while the brown<sup>1</sup> curve represents the average grades of the classified spectacles. The blue curve shows the ratio between the average grades of all the test sets and the maximal grades of the spoons test sets, in percentages. The learning sets in this experiment consisted of real scanned objects. The

test sets of spoons and spectacles are constant per experiment and comprised all the scanned objects, which is the entire database.

Fig. 16 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 brown curve, which gives the average grades of the spectacles, presents values of less than 0.1, while the black curve, which gives the average grades of the spoons, presents values that are higher than 0.1.

In an additional classification experiment, we used a training set of ten chairs where the ground support was implemented using a four-legged structure (see Fig. 17). 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 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 Fig. 17(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 employed the entire database in the cross-validation experiments. The learning (training) group represented 80% of the object class sets whereas the classification (or testing) group consisted of the rest of the database. Grades were computed during classification and the classified objects were evaluated as spectacles, forks, spoons, mugs, stools, tables, chairs, or airplanes. The grades were averaged and presented as textured bars in a graph, shown in Fig. 18. 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 Fig. 18, 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 and accuracy of classification

We performed the ROC and accuracy experiments on synthetic 3D models as well as on range images. In this section, the first set of experiments employed synthetic 3D models of 20 mugs and 20 spoons (Fig. 19 and Section 4.3.1). The second set of experiments was performed on the real range images of the entire database (Fig. 20 and Section 4.3.2).

We show several measurements of classification in terms of receiver operating characteristic curves (ROC) [10] as well as accuracy tests. In all the graph images, on the abscissas, we show the classification grade thresholds.

<sup>1</sup> For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.

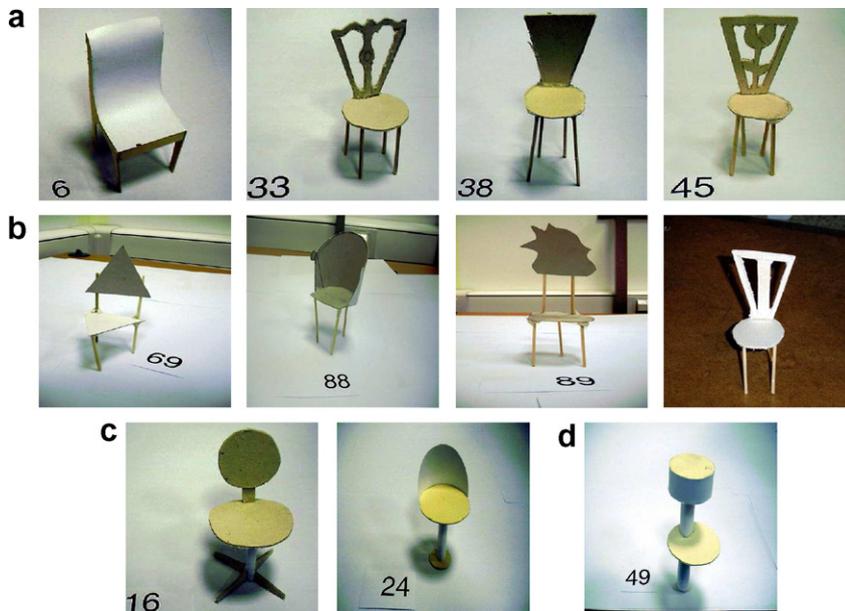


Fig. 17. Testing classification accuracy. (a) A subset of the training set images. (b) Three-legged chairs used in the experiment. All of them were correctly classified. (c and d) One-legged chairs that were used in the experiment. While the chairs in (c) were correctly classified, the scheme could not cope with the object in (d), due to its unusual back-support implemented by a blob.

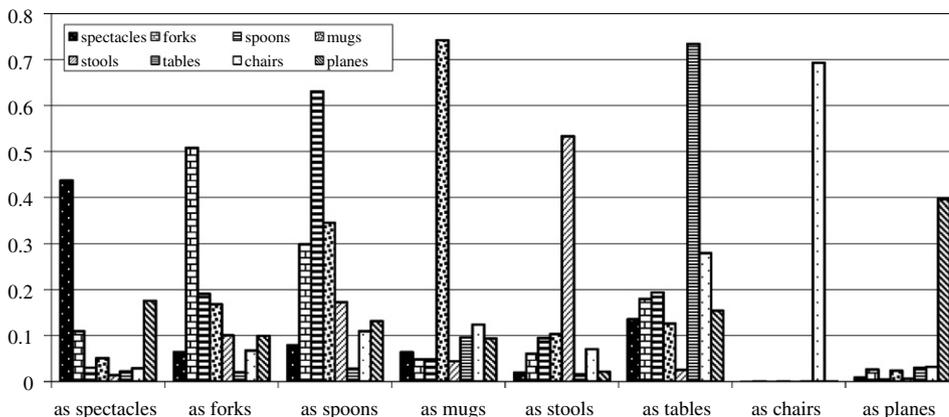


Fig. 18. Cross-validation on the whole database, which includes spectacles, forks, spoons, mugs, stools, tables, chairs, and airplanes.

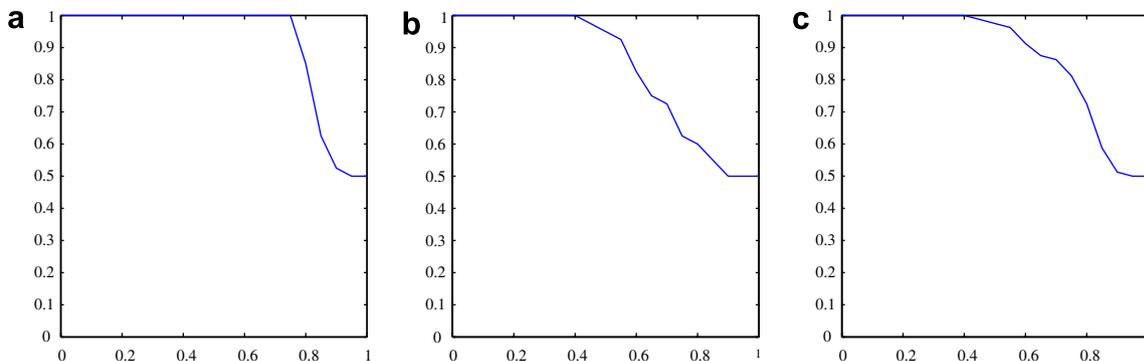


Fig. 19. Experiments on synthetic database that consists of mugs and stools. Synthetic 3D models of mugs and stools are employed. In experiments (a) and (b) we considered classifiers that select the maximum grades for the different components and check it versus thresholds. In (c) we built a composite classifier based on (a) and (b). We present the classification grade threshold on the abscissa. (a) The mug classifier's accuracy. (b) The stool classifier's accuracy. (c) Overall accuracy.

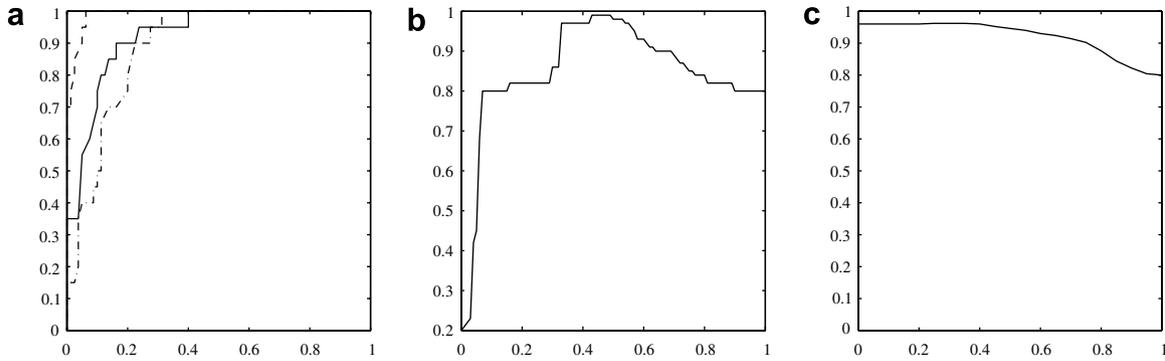


Fig. 20. Experiments on real range image data. The whole database was used. We show the classification grade threshold on the abscissa. (a) Three receiver operating characteristics for stools (the uppermost curve, dashed line), forks (the middle curve, normal line), and spectacles (the lowest curve, dot and dashed line). The individual classifiers used here do not perform a maximum on components. (b) The stool classifier's accuracy, which performs a maximum on components. (c) Overall accuracy.

#### 4.3.1. ROC and accuracy on synthetic data

Consider a synthetic database of mugs and stools. Here, 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. In Fig. 19(a), we show the accuracy of the mug classifier when working on the synthetic database. In a similar way, we defined a stools classifier. In Fig. 19(b), we show the accuracy of the stools classifier when working on the synthetic database. In Fig. 19(c), we show the accuracy of a combined classifier, that makes its decisions by selecting the maximum on the components of the vector grades. This classifier targets both mugs and stools and is tested on the same synthetic database.

We next built classifiers that work on the components of the vector matching grades. (These classifiers do not make their decisions by selecting the maximum on the components but by comparing the components of the vector 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 and accuracy on real range image data

In the real range image experiments, we first considered classifiers that work on the components of the vector matching grades (These classifiers do not make their decisions by selecting the maximum on components but by comparing grades to thresholds.). In Fig. 20(a), we show the superimposed ROC curves of stools, forks, and spoons.

Next, we considered 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 Fig. 20(b), we show the classifier's accuracy for stools versus other objects in the database.

The classification scheme should be as general as possible whenever the accuracy of classification schemes is being tested. Here, for clarity, we will refer to the classifiers described in the previous paragraph (and corresponding to Fig. 20(b)) as sub-classifiers. We built a combined classi-

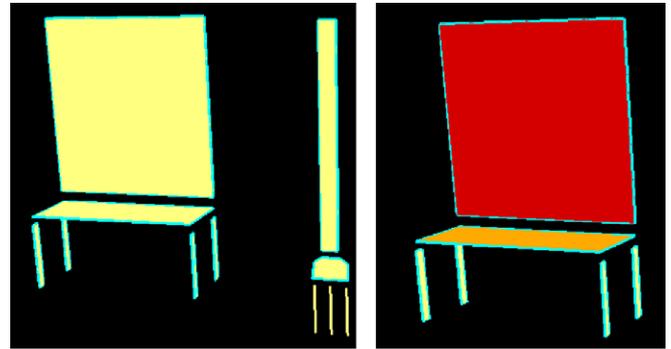


Fig. 21. A cluttered scene of a chair and a synthetically enlarged fork (left). The chair is correctly classified (right).

fier from the sub-classifiers described in the previous paragraph. This combined classifier calls the sub-classifiers on the components, selects the component grades that provided classification (those that are higher than predefined thresholds), and computes the maximum on the components. The maximum defines the classification result of the combined classifier. The combined classifier was tested on the whole database, which consists of five classes. In Fig. 20(c), unlike in Fig. 20(a) and (b), we show the accuracy of this combined classifier.

#### 4.4. Cluttered scenes

We performed the cluttered scene 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 Fig. 21). The second set is performed on the real range images of the entire database (see Fig. 22).

##### 4.4.1. Experiments on synthetic data

We built 33 synthetic scenes from 3D models of chairs, forks, and spoons. In Fig. 21, we show the classification results for the chair in a cluttered environment consisting

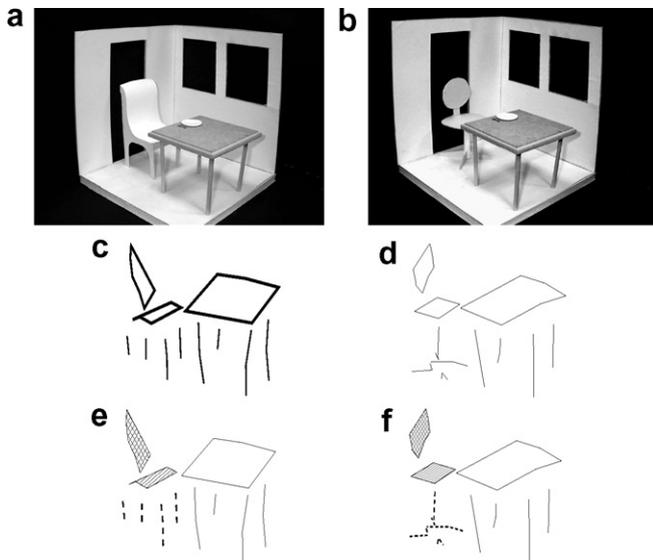


Fig. 22. Two cluttered scenes of a chair and a table in a room. (a) and (b) represent two examples of the first type of image. (c and d) The results of segmentation of the scenes in (a) and (b), respectively. The primitive parts sticks and plates are shown as they are detected and modelled in the segmentation phase. (e and f) The results of classifying the scenes in (a) and (b), respectively. (e and f) The resulting functional parts—the back support, seat, and the ground support—with different textures.

of the 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 Fig. 22 for examples of the first type). 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 these experiments was to demonstrate that our scheme has the ability to generalize to complex objects. We tried to recognize dining rooms and bedrooms. A room is a class with many subclasses; in other words, it has many specializations.

We used six rooms that contained a chair and a table and we characterized them as dining rooms (see

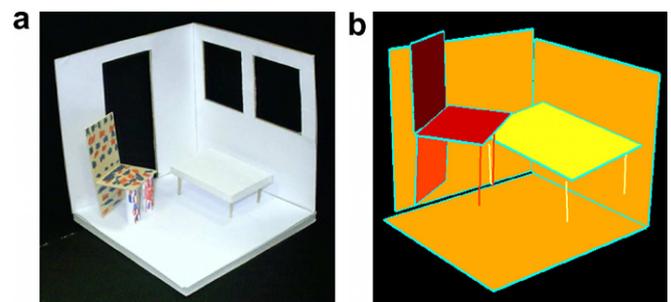


Fig. 24. Classification of a bedroom. By definition, a bedroom includes at least a bed. In this example, we show the classification of bedrooms versus dining rooms. (a) A digital photo of a room with a chair and a bed. The magenta color represents unselected parts during the classification process. The image in (b) represents the classification of the room as a bedroom, after the chair and the bed are selected (and colored) for analysis (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.).

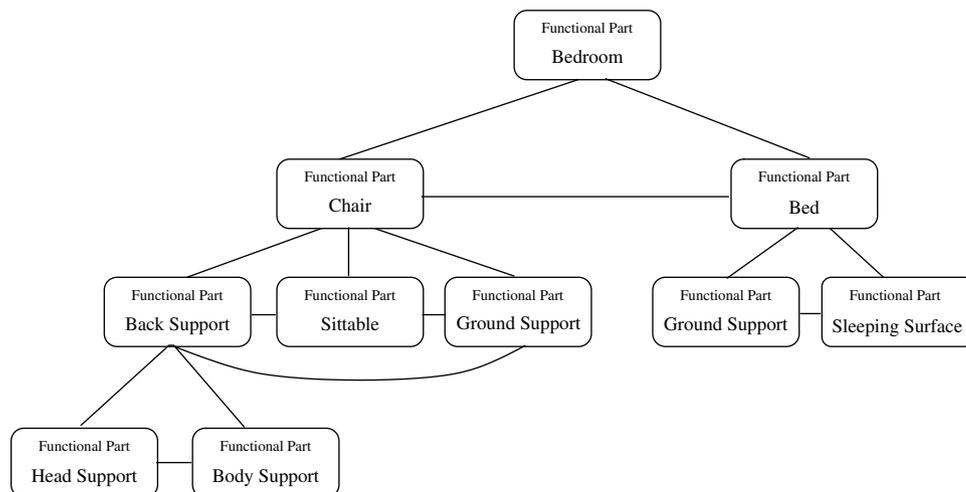


Fig. 23. Illustration of an *mlh*(room) representation with height four.

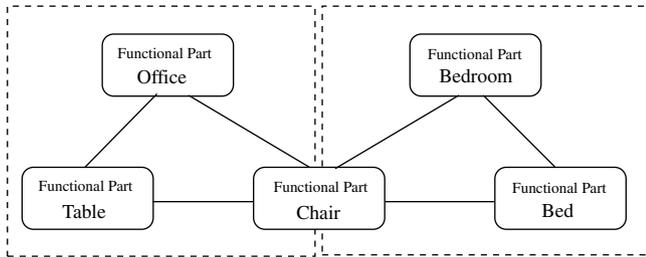


Fig. 25. Illustration of reusable function-based learning.

Fig. 15(a)). In addition, we used six rooms with a chair and a bed. We referred to these six rooms as bedrooms (see Fig. 15(b)). Note that the use of the chair in a room environment implies a multi-level hierarchy with height four (see Fig. 23). Here, the bed is an indication of a bedroom. In one dining room, the chair was not recognized due to heavy 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. Fig. 24 shows an image of a bedroom with classifications results. The classification of the chair and of the bed is shown Fig. 24.

When working in compound scenes one can take advantage of the reusable learning provided by function-based reasoning. For example, when learning dining rooms and bedrooms, the common denominator is learning chairs. This shared learning is illustrated in Fig. 25.

#### 4.6. Over-segmentation

In this section, we focus on the effects of over-segmentation to the classification results. We have learnt chairs as well as other several classes. Consider the chair in Fig. 26 (a), which is captured in a raw range image. We segmented and classified it as a chair. The results of the classification of this chair together with its functional sub-parts is shown using colors, each functional part being represented by a

unique color. Fig. 26(b) and (c) show the same chair when its back-support was manually modified. These versions were classified and the same color convention is used to underline the classification of functional components. While the functional parts of these chairs were still consistently classified, the grades of matching of these three input objects to class chair are significantly different. We report that we tested our scheme with different sets of geometric properties. When experimenting our scheme with different sets of geometric properties, the maximum matching grade among the three chairs is not consistent, i.e., we cannot point out a version that clearly is the best matching. However, our scheme classified the three objects as chairs consistently.

## 5. Conclusions

In this work, we have presented a novel function-based scheme for classification of 3D objects. The input consists of 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 1000 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. 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.

Some of our future work consists of enlarging the database of the test objects. Specifically, in future experiments, we intend to introduce additional categories of

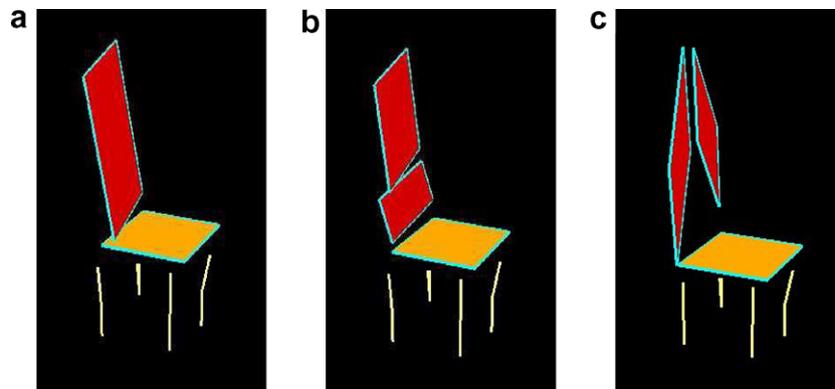


Fig. 26. Each color identifies a classified functional part. The chair in (a) is a real range image segmented. Figures (b) and (c) show the same chair, when its back-support was intentionally over-segmented in plates.

range image objects. In addition, we are going to use more accurate approximation models for primitive parts and use more elaborate models to describe these parts. However, it should be noted that the use of relatively coarse parts had no negative influence on recognition of difficult categories. 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.

### Acknowledgments

The authors thank Diana Soldea, Nela Gurevich, Emil Dan Kohn, Lev Ferdinkoif, Ohad Serfaty, Vitali Greenberg, Artyom Sharov, Yanai Shpinner, Fady Massarwi, Ebraheem Kassem, Adir Abraham, Ronen Keidar, and Yekutiel Pekar for their support.

### Appendix A. Geometric properties

See Tables A.1 and A.2.

Table A.1

Geometric properties for primitive and functional parts

CircularRegularityByAngle	CircularRegularityByRadius
FPartConnectivity	LinearRegularity
Stability	LinearSymmetry
CircularSymmetry	RegularSurface
RegularVolume	BBoxSurface
PPartsMajorAxesStandardDeviation	InertiaMomentsIxx
InertiaMomentsIxy	InertiaMomentsIxz
InertiaMomentsIyy	InertiaMomentsIyz
InertiaMomentsIzz	InertiaEigenValue1
InertiaEigenValue2	InertiaEigenValue3
BBoxVolume	RelativeVolumeBBoxVolume
RelativeSurfaceBBoxSurface	COMDistanceFromBBoxCenter
CenterOfMassWithinBBox	FPartOrientation

The full definitions of these properties can be found in [34].

Table A.2

Geometric properties for associations

CrucialConnectivity
CrucialStability
RelativeFPartCenterOfMass
RelativeFPartVolume
RelativeSurface
RelativeFPartBBoxVolume
RelativeFPartBBoxSurface
MassCentersRelativeDirection
MassCentersRelativeDistance
NumberOfPPartConnections
TypeOfConnection
ConnectionGeometricDescription
BBoxConnection

The full definitions of these properties can be found in [34].

### References

- [1] R. Bajcsy, F. Solina, Three dimensional object representation revisited, in: IEEE Proceedings of the International Conference on Computer Vision, 1987, pp. 231–240.
- [2] R. Bergevin, M.D. Levine, Generic object recognition: building and matching coarse descriptions from line drawing, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1) (1993) 19–36.
- [3] E. Bicipi, R. St. Amant, Reasoning About the Functionality of Tools and Physical Artifacts, NC State University, Technical Report, No. 22, 2003, <ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc\_anon/tech/2003/TR-2003-22.pdf>.
- [4] M. Brady, P.E. Age, D.J. Braunegg, J. Connell II, The Mechanic's Mate, in: Proceedings of the Sixth European Conference on Artificial Intelligence, 1984, pp. 79–94.
- [5] M. Brand, Physics-based visual understanding, Computer Vision and Image Understanding 65 (2) (1997) 192–205.
- [6] R.A. Brooks, Model-based three-dimensional interpretations of two-dimensional images, IEEE Transactions on Pattern Analysis and Machine Intelligence 5 (2) (1983) 140–149.
- [7] R.A. Brooks, R. Greiner, T.O. Binford, The ACRONYM model-based vision system, in: Proceedings of the International Joint Conference on Artificial Intelligence, 1979, pp. 105–113.
- [8] M.D. Buhmann, M.J. Ablowitz, Radial Basis Functions: Theory and Implementations, Cambridge University, Cambridge, 2003, ISBN 0-521-63338-9.
- [9] E. Davis, The Naive Physics Perplex, AI Magazine, 1998.
- [10] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Second ed., John Wiley and Sons, New York, 2001.
- [11] M. DiManzo, E. Trucco, F. Giunchiglia, F. Ricci, FUR: understanding functional reasoning, International Journal of Intelligent Systems 4 (1989) 431–457.
- [12] C. Dorai, A.K. Jain, COSMOS—a representation scheme for 3D free-form objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (10) (1997) 1115–1130.
- [13] T.J. Fan, G.G. Medioni, R. Nevatia, 3-D Object Recognition Using Surface Descriptions, Image Understanding Workshop, pp. 383–397, 1988.
- [14] G. Farin, Curves and Surfaces for CAGD—A Practical Guide, fourth ed., Academic Press Inc, New York, 1996.
- [15] G. Froimovich, E. Rivlin, I. Shimshoni, Object Classification by Functional Parts, in: Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission—3DPVT'02, 19–21 June, 2002, pp. 648–655.
- [16] J. Gibson, The Ecological Approach to Visual Perception, Houghton Mifflin, Boston, 1979.
- [17] K. Green, D. Eggert, L. Stark, K. Bowyer, Generic recognition of articulated objects through reasoning about potential function, Computer Vision and Image Understanding 62 (2) (1995) 177–193.
- [18] W.E.L. Grimson, The combinatorics of heuristic search termination for object recognition in cluttered environments, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (9) (1991) 383–397.
- [19] M. Hebert, T. Kanade, I. Kweon, 3-D vision techniques for autonomous vehicles, Analysis and Interpretation of Range Images (1990) 273–337.
- [20] J. Hodges, Functional and physical object characteristics and object recognition in improvisation, Computer Vision and Image Understanding 62 (2) (1995) 147–163.
- [21] R. Hoffman, A.K. Jain, Evidence-based recognition of 3-D Objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (6) (1988) 783–802.
- [22] A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D.B. Goldgof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, R.B. Fischer, An experimental comparison of range image segmentation algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (7) (1996) 673–689.

- [23] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: *IEEE Proceedings of the Conference on Computer Vision and Pattern Recognition*, vol 1, 17–21 October 2005, pp. 604–610.
- [24] J.F. Kenney, E.S. Keeping, *Mathematics of Statistics Part 1*, third ed., van Nostrand, Princeton, NJ, 1962, pp. 77–80.
- [25] Y. Keselman, S. Dickinson, Generic model abstraction from examples, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (7) (2005) 1141–1156.
- [26] D.E. Knuth, *The Art of Computer Programming*, vol 3, first ed., Addison-Wesley, Reading, MA, 1973.
- [27] D.J. Kriegman, T.O. Binford, T. Sumanaweera, Generic Models for Robot Navigation, *Image Understanding Workshop*, 1988, pp. 453–460.
- [28] W.-J. Li, T. Lee, Object recognition and articulated object learning by accumulative hopfield matching, *Pattern Recognition* 35 (2002) 1933–1948.
- [29] T.M. Mitchell, *Machine Learning*, McGraw-Hill, New York, 1997.
- [30] M. Minsky, Commonsense-based interfaces, *Communications of the ACM* 43 (8) (2000) 67–73.
- [31] M. Minsky, A Framework for Representing Knowledge, MIT-AI Laboratory Memo 306, June, 1974. Reprinted in *The Psychology of Computer Vision*, P. Winston (Ed.), McGraw-Hill, New York, 1975.
- [32] L. Morgenstern, Mid-sized axiomatizations of commonsense problems: a case study in egg cracking, *Studia Logica* 67 (2001) 333–384.
- [33] J. Mundy, A. Liu, N. Pillow, A. Zisserman, S. Abdallah, S. Utcke, S. Nayar, C. Rothwell, An experimental comparison of appearance and geometric model based recognition, in: *Proceedings of Object Representation in Computer Vision II*, 1996.
- [34] <[www.cs.technion.ac.il/~mpechuk/ProjectOCLS/index.html](http://www.cs.technion.ac.il/~mpechuk/ProjectOCLS/index.html)>.
- [35] A. Opelt, A. Pinz, Andrew Zisserman, Fusing Shape and Appearance Information for Object Category Detection, *Proceedings of the British Machine Vision Conference* (2006).
- [36] A.P. Pentland, Automatic extraction of deformable part models, *International Journal of Computer Vision* 4 (2) (1990) 107–126.
- [37] P. Peursum, S. Venkatesh, G.A.W. West, H.H. Bui, Using interaction signatures to find and label chairs and floors, *IEEE Pervasive Computing* 3 (4) (2004) 58–65.
- [38] E. Rivlin, S.J. Dickinson, A. Rosenfeld, Recognition by functional parts, *Computer Vision and Image Understanding* 62 (2) (1995) 164–176.
- [39] E. Rosch, Cognition and categorization, in: E. Rosch, B. Lloyd (Eds.), Erlbaum, Hillsdale, NJ, 1978.
- [40] R.St. Amant, A.B. Wood, Tool use for autonomous agents, in: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2005, pp. 184–189.
- [41] L. Stark, K.W. Bowyer, Function-based generic recognition for multiple object categories, *Computer Vision, Graphics, and Image Processing* 59 (1) (1994) 1–21.
- [42] L. Stark, A.W. Hoover, D.B. Goldgof, K.W. Bowyer, Function-based recognition from incomplete knowledge of shape, *Workshop on Qualitative Vision*, 1993, pp. 11–22.
- [43] M.A. Sutton, L. Stark, K.W. Bowyer, GRUFF-3: generalizing the domain of a function-based recognition system, *Pattern Recognition* 27 (12) (1994) 1743–1766.
- [44] M.A. Sutton, L. Stark, K. Hughes, Exploiting context in function-based reasoning, *Lecture Notes in Computer Science* 2238 (2002) 357–373.
- [46] B. Triggs, P.F. McLauchlan, R.I. Hartley, A.W. Fitzgibbon, Bundle Adjustment A Modern Synthesis, *Vision Algorithms' 99*, in: B. Triggs, A. Zisserman, R. Szeliski (Eds.), *Lecture Notes in Computer Science*, 1883 (2000) 298–372.
- [47] A. Turing, Computing machinery and intelligence, *Mind* 59 (1950) 433–460.
- [48] S. Ullman, *High-Level Vision, Object Recognition and Visual Cognition*, The MIT Press, Cambridge, MA, 1995.
- [49] L.M. Vaina, M.C. Jaulent, Object structure and action requirements: a compatibility model for functional recognition, *International Journal of Intelligent Systems* 6 (1991) 313–336.
- [50] B.C. Vemuri, J.K. Aggarwal, Representation and recognition of objects from dense range maps, *IEEE Transactions on Circuits and Systems* 34 (11) (1987) 1351–1363.
- [51] P.H. Winston, T.O. Binford, B. Katz, M. Lowry, Learning Physical Descriptions from Functional Descriptions, *Proceedings of the National Conference on Artificial Intelligence* (1983) 433–439.
- [52] K. Woods, D. Cook, L. Hall, K. Bowyer, L. Stark, Learning membership functions in a function-based object recognition system, *Journal of Artificial Intelligence Research* 3 (1995) 177–222.