Function-Based Classification from 3D Data via Generic and Symbolic Models

Michael Pechuk, Octavian Soldea, and Ehud Rivlin
Technion - Israel Institute of Technology
Computer Science Department, Artificial Intelligence Laboratory
Haifa, Israel 32000
e-mail:{mpechuk,octavian,ehudr}@cs.technion.ac.il

Abstract

We propose a novel scheme for function-based classification of objects in 3D images. The classification process calls for constructing a generic multi-level hierarchical description of object classes in terms of functional components. Functionality is derived from a large set of geometric attributes and relationships between object parts. Initially, 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 and functional parts at the same level in the hierarchy are labeled as well. Then, the generic multi-level hierarchical description of object classes is built using the functionalities of a number of object instances. During classification, a search through a finite graph using a probabilistic fitness measure is performed to find the best assignment of object parts to the functional structures of each class. An object is assigned to a class providing the highest fitness value. The scheme does not require a-priori knowledge about any class. We tested the proposed scheme on a database of about one thousand different 3D objects. The results show high accuracy in classification.

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). We propose a new scheme that is able to classify objects from range images.

There is a plethora of object recognition approaches. The first approaches dealt with single object models in the input. They were based on geometric modelling. Later on, the parameterized geometric modelling was introduced (Ullman 1995). A fundamentally different approach was introduced by Gibson (Gibson 1979), who considered that the human mind classifies objects according to usage, i.e., by the functions that an object may fulfil.

The first systems using function-based classification were (Winston *et al.* 1983) and (DiManzo *et al.* 1989). An impressive number of results in the function-based classification field were demonstrated with the GRUFF and OMLET systems (Stark & Bowyer 1994; Sutton, Stark, & Bowyer 1994; Woods *et al.* 1995). The following issues were addressed: reuse of a limited set of knowledge primitives in defining an expanded domain of competence, computing an association measure for the appropriateness of a shape that can be compared across categories so that different interpretations of a shape can be rank ordered, and learning membership functions from 3D objects. The learning phase aims augmenting functions that are defined in a human-driven preprocessing. To the best of our knowledge, GRUFF and OMLET were tested on raw images that included chairs that were artificially built from boxes.

In (Rivlin, Dickinson, & Rosenfeld 1995), the authors presented a theory of function-based recognition that is a natural extension of the part-based shape recognition approach. Following (Rivlin, Dickinson, & Rosenfeld 1995), in (Froimovich, Rivlin, & Shimshoni 2002), the authors proposed a system for function-based classification. The classification approach, performed on range images of *real* 3D objects, assumes a-priori knowledge of the objects.

We propose a novel scheme for function-based classification of objects using 3D images. The classification process calls for constructing a generic multi-level hierarchical description of object classes 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 and functional parts at the same level in the hierarchy are labeled as well. Then, the multi-level hierarchy, which is a probabilistic model of an object class, is defined using histograms built from the functionalities of a number of object instances. Our scheme is able to automatically build the description of any new object class from labeled examples.

During classification, a search through a finite graph using a probabilistic fitness measure is performed to find the best assignment of parts of an object to the functional structures of each class. An object is assigned to a class providing the highest fitness value.

Function-based approaches offer the advantage of granular learning. This means that functional parts that have the same geometry and are shared between different classes do not have to be learnt with each new class. This fact enables marked acceleration of the learning phase. To the best of our knowledge, no such sharing was used nor described in existent systems.

We tested the proposed scheme on a database of about one thousand different 3D objects. As far as we have been able to determine, no other classification (or recognition) scheme has yet been tested on hundreds of range images of real objects captured in range images. In so far as we know, this is the first scheme that performs function-based classification that involves a learning phase and does not require a-priori knowledge about any class.

Learning and Classifying Functionalities

The proposed scheme consists of two phases: a learning phase and a classification phase. Each of these phases receives as input segmented images. The objects are segmented and labeled into constituents: primitive and functional parts. The primitive parts (also known in the literature as geons (Biderman 1987), (Dickinson *et al.* 1997)) that we consider are sticks, plates, and blobs. A functional part is defined as an object part that could provide a certain function, and usually comprises several primitive parts; for example, a ground support of a chair consists of four parallel stick primitive parts. Note that (Biderman 1987) and (Dickinson *et al.* 1997) stated that thousands of objects can be mapped to a low number of primitive parts. The immense number of objects in nature is the result of the combinatorial number of interrelationships between the primitive parts.

In the learning phase, several instances (objects) of a class are input. The learning phase computes the values of the geometric attributes of the constituents and the relationships between them. Once the learning phase is finished, the generic representation has accumulated enough information for classification - the next phase.

Segmentation

The learning as well as the classification phases receive as input primitive parts as detected from a segmentation algorithm, which is a variation on UE (Hoover *et al.* 1996). The segmentation provides representation models for the primitive parts. We do not elaborate on the details of the implemented segmentation technique due to space limitations.

Multi-Level Hierarchy Functional Structure

The classification process comprises an analysis both of the detected primitive parts and the relationships that exist among them. We call the relationships between primitive parts primitive-to-primitive connections. We have generalized the mechanism of decomposition into parts and primitive-to-primitive connections into a multi-level approach in the following sense. Each primitive part or group of primitive parts and the primitive-to-primitive connections among them that can fulfil a certain functional task are classified as a functional part. Further, several functional parts and the relationships among them can define a functional task and can form a higher level functional part. The proposed hierarchy can be as complex as one wishes. This approach is known in the literature as recognition/classification by functional parts.

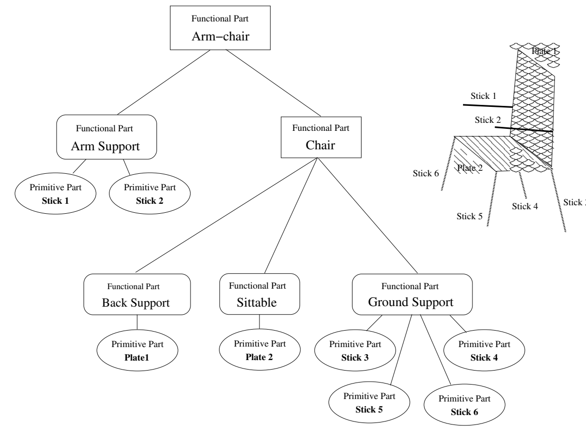


Figure 1: The multi-level hierarchy functional structure. $F\left(\text{Arm-chair}\right) = \left\{\text{Arm Support}, \text{Chair}\right\}$. The arms represent a simple functional part with the functionality of supporting the arms, while the chair is a high level functional part due to the fact that it describes a more complex functionality.

A relationship between a pair of functional parts is called a functional-to-functional connection. Whenever it is clear from the context, we use the term connection to denote a primitive-to-primitive or functional-to-functional connection. Each level in the functional hierarchy has a clique structure and each pair of functional parts (in the clique) are characterized by a relationship expressed in terms of geometric attributes. For example, in Fig. 1, each pair of functional parts ”Back Support”, ”Sittable”, and ”Ground Support” are connected, thus forming a clique. Note that these three nodes have the common ancestor ”Chair”.

The multi-level hierarchy functional structure of an object class is implemented by a layered tree structure. For any functional part f, define $P\left(f\right)$ and $F\left(f\right)$ the set of immediate primitive or functional constituents of f and $C\left(f\right)$ the set of connections between the elements of $P\left(f\right) \bigcup F\left(f\right)$; see Fig. 1. Note that only one of $P\left(f\right)$ or $F\left(f\right)$ is not empty for any functionality f.

For any symbolic primitive part, functional part, or connection a, we associate $GP\left(a\right)$, a set of geometric attributes. If a is a primitive or a functional part then $GP\left(a\right)$ includes, among other properties, inertia moments, stability, and regularity. If a is a connection, $GP\left(a\right)$ includes, among other properties, ratio of volumes and context-based stability. (We define the context-based stability attribute of a connection being the stability of the ensemble of the parts related to this connection.) The full description of the geometric attributes we have considered is relatively large and can be found at www.cs.technion.ac.il/∼mpechuk/ProjectOCLS/index.html.

Each geometric attribute is associated with a histogram of measured values. The histograms are built employing B-spline functions (Farin 1996) (we use uniform knot sequences and employ cubic B-splines in this work) . These B-spline functions allow the computation of hypotheses with relationship information about the primitive/functional parts. The multi-level hierarchy of each learnt class is stored in a database.

Consider a multi-level hierarchy and let P and F be the set of all the symbolic primitives and functional parts, respectively, the hierarchy includes. Define $A = \{(a, g) \mid a \in F \bigcup P, g \in GP(a)\}$ and $B = \bigcup_{f \in F} \{(c, g) \mid c \in C(f), g \in GP(c)\}$. Then, the multi-level hierarchy of a functionality f induces a function $\mathcal{H}_f : A \bigcup B \rightarrow H$, where $H = \{h \mid h : R \rightarrow [0..1]\}$ is the set of all (normalized) histograms that can be implemented as B-spline functions (see next section) .

Learning Functionalities

Fig. 2 (upper part) shows the flow of the learning phase of our scheme. The input of the learning phase is a set of functional labeled objects. 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 pre-defined geometrical attributes. Further, these attributes are subject to an RBF-like (radial-based function) learning (Mitchell 1997). We mention that, unlike (Mitchell 1997), we chosen to implement histograms via B-spline functions and not Gaussian ones due to a slightly better accuracy offered by B-spline functions. The accuracy issue was tested on some preliminary experiments that we do not provide here due to space limitations.

In the learning phase, the scheme builds histograms for geometric properties of the functional parts as well as for the connections between the functional parts. The continuous domain of measured values for geometric properties is approximated by discrete accumulation values which are provided as the coefficients to B-spline functions that implement the histograms. The scalar coefficients are normalized such that the maximum coefficient equals 1.0. Note that this process is automatic, does not require other operator intervention but labeling.

Once the geometric properties of the primitive parts are pre-evaluated, the reasoning is conducted via histograms based grades, to be defined in the next section. For each functional part, the set of histograms of its constituents, functional (sub)-parts and connections, represents the signature of the functional part.

Function-based approaches offer the advantage of granular learning, that is, functional parts that have the same geometry and are shared between different classes do not have to be learnt with each new class. We exploit this advantage in speeding up the scheme for learning new objects; that is, we design the learning sequences from objects with functional parts having different geometry. Repetition of geometrically similar functional parts is not necessary. Nevertheless, it can be used to dictate a bias in classification.

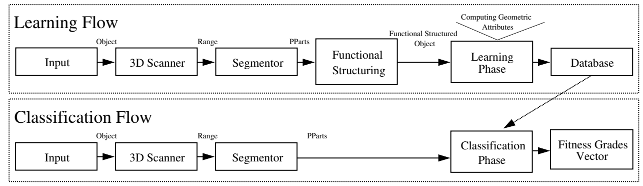


Figure 2: Learning and classification flows.

Classification

Fig. 2 (lower part) 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 multi-level hierarchy provided by the learning phase. The classification phase computes a vector of grades that describes how an object offers class functionalities. Each element of the vector represents a grade relative to one class.

The class with the highest fitness grade is chosen as the best match found by our scheme. For each one of the learnt classes the scheme tries to find the best realization for the multi-level hierarchy functional structure out of the given set of primitive parts and the connections between them. (By realization we mean a partition for which we have evaluated its grades' vector of fitness.) A fitness grade is then computed for each such realization. Thus, we reduce the problem of classifying a new object to the problem of finding the multi-level hierarchy functional structure realization with the highest fitness grade, which in fact relies on computing a partition. More specifically, we define a partition as a set of primitive parts sub-sets, each set representing a functional element in the multi-level hierarchy. The following subsections describe the fitness grade computation process and several partitioning algorithms.

Fitness Grades Computation Assume we want to evaluate the fitness grade for functionality f and FH is a multi-level hierarchy computed for f. Let P_{in} an input set of primitive parts which are to be partitioned in the functional parts of FH in order to recover f. Let $\overline{P}$ be the set of all hypothetic partitions of P_{in} in a FH structure and for any $s \in F(f) \bigcup P(f)$ let $\overline{p}(s)$ be the sub-set of the partition $\overline{p}$ relative to s. Define

$$grade(f, \overline{p}) = \sum_{g \in GP(f)} \mathcal{H}_f(f, g)(\overline{p}) \prod_{s \in \left(F(f) \bigcup C(f)\right)} grade(s, \overline{p}(s))$$

if $P(f)$ is empty and

$$grade(f, \overline{p}) = \left(\sum_{g \in GP(f)} \mathcal{H}_f(f, g)(\overline{p}) \right) \left(\sum_{s \in \left(P(f) \bigcup C(f)\right)} w(s, g)\, grade(s, \overline{p}(s)) \right)$$

otherwise. Here, $w(s, g)$ is a weight function that is proportional with the standard deviation (Kenney & Keeping 1962) of the histogram function itself corresponding to s and g (not the histogram's range) and $\mathcal{H}_f(s, g)(\overline{p})$

means the value of the geometric property histogram for s implemented following partition $\overline{p}$. If p is a primitive part then the fitness grade $grade\,(p, \overline{p}\,(p)) = \sum_{g \in GP(p)} \mathcal{H}_f\,(f, g)\,(\overline{p}\,(p))$. If c is a connection, then

$$grade\,(c, \overline{p}) = \sum_{g \in GP(c)} w\,(c, g)\,\mathcal{H}_f\,(c, g)\,(\overline{p})\,.$$

Moreover, for any functionality f, the fitness grade is defined as

$$grade\,(f) = \max_{\overline{p} \in \overline{P}} grade\,(f, \overline{p})\,.$$

The weights are used to emphasize the geometric attributes that have a higher potential to characterize a specific class. We assume that this potential is higher in geometric attributes that have histograms with peaks than in ones that have constant value. In order to determine the best geometric attributes we compute the weights as standard deviations (Kenney & Keeping 1962) of the histograms values. Note that the weights mechanism can eliminate unnecessary connections in cliques.

The classification phase is a search and validation like algorithm over a finite graph of partitions. 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 serve as its seat, its back support, and one of its legs (as part of its ground support). This motivates our interest in designing algorithms for partitioning.

Matching Partitions to Functionalities We define the matching of partitions to functionalities as a search in a finite graph problem. The nodes and edges of the search graph are described below.

A function that associates a list of primitive parts to each functional part defines a partition. In addition, we define a "non-partitioned" set that contains all not matched primitive parts. Each node of the graph contains the above sets. We assume that the input 3D model consists of n primitive parts whereas the analyzed class consists of m functional parts. The search space is a graph with $(m+1)^n$ nodes. The first state of the graph is always the "empty" state, i.e., all the primitive parts are located in the "non-partitioned" section. The children's generation function takes a state of level $k-1$ in the graph and generates all possible realizations for the k-th functional part from the "non-partitioned" primitive parts set. It assumes that the previous $k-1$ functional parts are already realized. Thus, the last level contains all possible partitions of the object. The goal is to find the state with the highest fitness grade.

We used a heuristic search with a pruning branch-and-bound approach. Define the partial fitness grade of a primitive part, a functional part, or connection a relative to partition $\overline{p}$ be

$$partial\,(f, n) = \begin{cases} grade\,(f, \overline{p}) & \text{if } f \text{ is assigned} \\ 1 & \text{otherwise} \end{cases}, \quad (1)$$

where n is a node in the search graph. For searching purposes we use partial fitness grades. From (1), it follows that when the search reaches a leaf, the partial grade equals the fitness grade.

Following (Grimson 1991), the algorithm searches the states' graph starting from the "empty" state. The algorithm searches for the partition that has the highest fitness grade, which represents the classification result. We have, however, changed the identification of primitive parts into comparing signatures of functional parts with geometrical attributes of the functional part candidates.

In our tests we used an exhaustive search algorithm as well as a genetic search algorithm. Although the genetic algorithm erred in 10% of the classifications, it has the advantage that is time bounded.

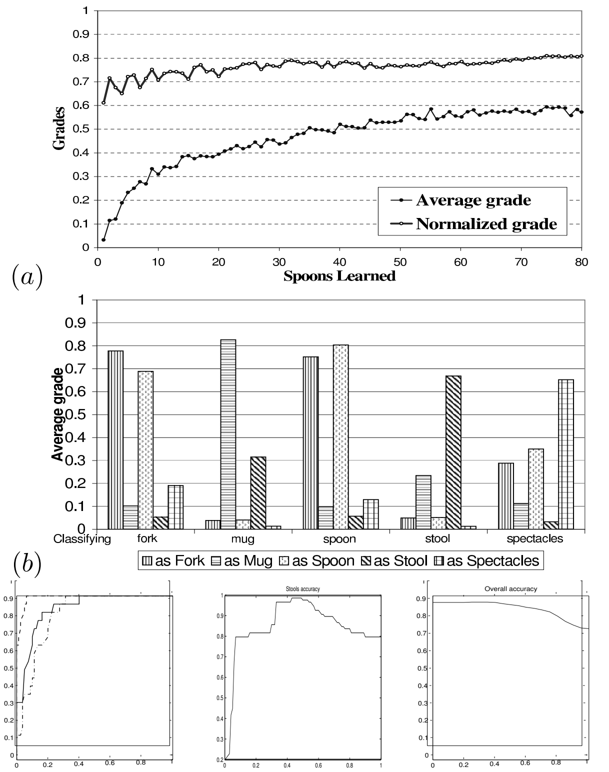
Experiments

We tested our scheme on a database comprising synthetic models of 200 forks, 216 spoons, 200 stools, and 200 spectacles. We also tested our scheme on a database comprising 100 forks, 100 spoons, 97 chairs, and 100 spectacles, of real range images. Partial sets of the forks, spoons, chairs and the spectacles, that we used in experiments, are shown in Fig. 5 and 6. The objects in the range images were captured with a Cyberware range scanner (http://www.cyberware.com).

We performed four types of experiments aiming to check model strength, cross validation, receiver operating characteristic (ROC) and accuracy, and classification in cluttered scenes. In all the tests, the learning phase was performed on images that contain only one object. In model strength checking, cross validation, and ROC and accuracy tests, we used the proposed scheme to classify objects from 3D images that contain only one object. In the cluttered scene tests, partial views of several objects were used.

Model Strength Checking In the model strength checking experiment, two sets of objects were used: a learning set and a test set. The graph in Fig. 3 (a) shows the test set average grades as a function of the size of the training set. Here, the lowest curve represents the average of the grades of the classified objects and the higher curve shows the percentage of the test set's average grade from the maximal grades in the test set. In the experiment shown in Fig. 3 (a) the learning sets consisted of real scanned objects. The set used for classification is constant per experiment and comprised all the scanned objects.

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 consists of the rest of the data base. The result of the classification consists of computing grades: the classified objects are evaluated as fork, mug, spoon, stool, and spectacles. The grades are averaged and presented as textured bars in a graph, that is shown in Fig. 3 (b). The graph comprises five groups of five object class grades. For example, the first group relates to forks that were classified as forks, mugs, spoons, stools, or spectacles in this order. In Fig. 3 (b), the forks, the spoons, and the spectacles were range images. The tests on real data revealed that, although we can differentiate between forks and spoons, the degree



(c) (d) (e)

Figure 3: Experiments: (a) Learning and classifying real spoons. (b) Cross validation on the whole data base, which includes forks, mugs, spoons, stools, and spectacles. (c) A ROC on the whole data base for stools (the uppermost curve), forks, and spectacles (the lowest curve). (d) The accuracy of stools versus the rest of the data base. (e) Overall accuracy.

of differentiation is quite low. This is mainly due to the fact that spoons and forks have similar dimensions, forms, and functionality (both have handles and are graspable).

ROC and Accuracy Consider classifiers that work on components of the vector fitness grades (the classifiers do not perform maximum on components). In Fig. 3 (c), we show the ROC superimposed curves of stools, forks, and spoons. In Fig. 3 (d), we show the accuracy of the classifier of stools versus the data base. In Fig. 3 (e), unlike in Fig. 3 (c) and (d), we show the accuracy of the classifier that performs maximum on vector grades, targeting all five tested classes and the whole data base.

Cluttered Scenes In the cluttered scene tests, 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 of images included a chair and a table in a room while the second one included chairs and spoons (see Fig. 4 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 of the images from the first type and in all the images of the second type, our system correctly classified a valid chair.

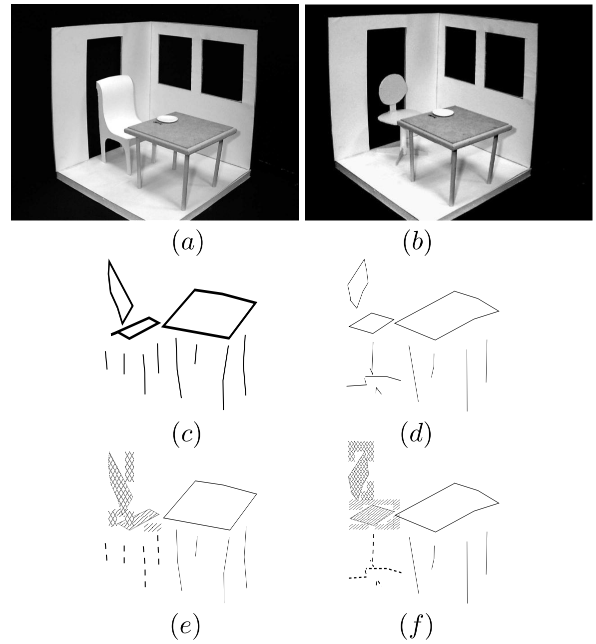


Figure 4: Two cluttered scenes of a chair and a table in a room. (a) and (b) represent two examples of the first type of image. Fig. (c) and (d) represent results of segmentation of the scenes in (a) and (b), respectively. The primitive parts sticks and plates are shown as they are detected and modelled in the segmentation phase. Fig. (e) and (f) represent results of classifications of 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.

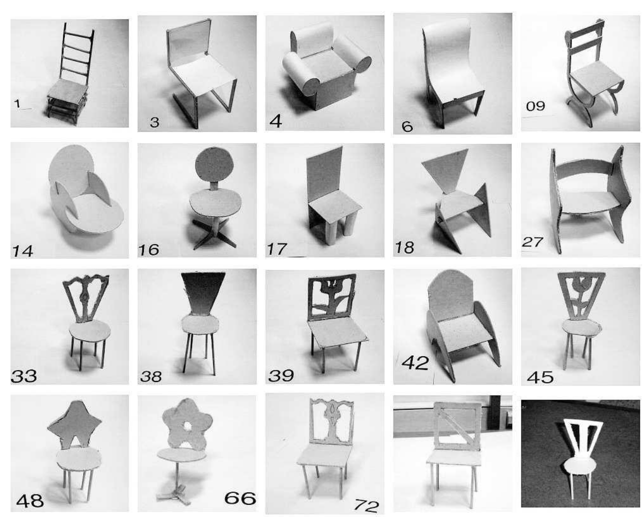


Figure 5: Images of some chairs used in the experiments.

Conclusions

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

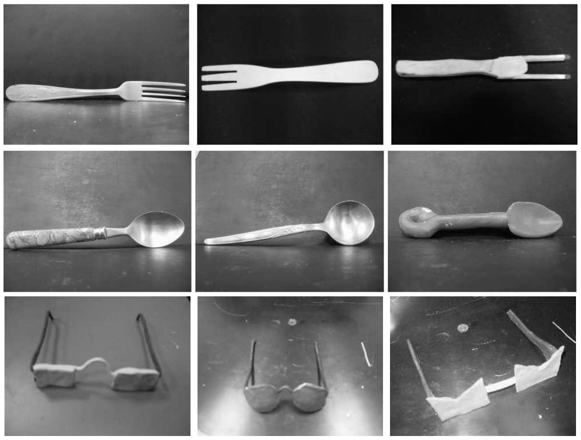


Figure 6: Images of some forks, spoons, and spectacles used in our experiments.

are full 3D descriptions of objects. The proposed scheme employs an object functional structure, consisting of a multi-level hierarchy of functional parts. The multi-level approach offers a higher degree of freedom for real object modelling as compared to classical systems and can be seen as a learning phase. The learning phase builds the multi-level hierarchy by analyzing primitive parts in the input images and their geometric attributes.

Our approach was tested on a database of about one thousand different 3D objects employing several algorithms for searching and pruning. To the best of our knowledge, no other classification (or recognition) scheme was tested on hundreds of range images of real objects captured in range images. The graphs show the quality achieved by our scheme. They also provide an insight into the dimensions of the learning sets that are required so as to reach a certain degree of classification accuracy. Our work appears to be the first scheme that performs function-based classification that involves a learning phase and does not require a-priori knowledge about any class.

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 proposed solution is clearly parallelizable. We believe that employing concurrent or parallel variants of the algorithms, and/or implementing the classification evaluation schemes on dedicated hardware, could greatly speed up the classification process.

Part of our future work consists of enlarging the data base of the testing objects. Specifically, we intend to introduce more categories of objects captured in range images in the experiments. In addition, we are going to use more expressive approximation models for primitive parts. The more exact the approximation models are, the more accurate the classification is expected to be.

Acknowledgments

The authors thank Nela Gurevich, Lev Ferdinskoif, Ohad Serfaty, Vitali Greenberg, Artyom Sharov, Yanai Shpinner, Fady Massarwi, Ebraheem Kassem, Leonid Raskin, Adir Abraham, Ronen Keidar, and Yekutiel Pekar for their support.

References

Biderman, I. 1987. *Recognition by Components: A Theory of Human Image Understanding*. MIT Press.

Dickinson, S.; Bergevin, R.; Biederman, I.; Eklundh, J.; Munck-Fairwood, R.; Jain, A.; and Pentland, A. 1997. Panel report: The potential of geons for generic 3-d object recognition. *IVC* 15(4):277–292.

DiManzo, M.; Trucco, E.; Giunchiglia, F.; and Ricci, F. 1989. Fur: Understanding functional reasoning. *International Journal of Intelligent Systems* 4:431–457.

Farin, G. 1996. *Curves and Surfaces for CAGD - A Practical Guide. Fourth Edition.* Academic Press Inc., Publishers New York.

Froimovich, G.; Rivlin, E.; and Shimshoni, I. 2002. Object classification by functional parts. *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission - 3DPVT'02* 648–655.

Gibson, J. 1979. *The Ecological Approach to Visual Perception*. Houghton Miffin, Boston.

Grimson, W. E. L. 1991. The combinatorics of heuristic search termination for object recognition in cluttered environments. *IEEE PAMI* 13(9):383–397.

Hoover, A.; Jean-Baptiste, G.; Jiang, X.; Flynn, P. J.; Bunke, H.; Goldgof, D. B.; Bowyer, K.; Eggert, D. W.; Fitzgibbon, A.; and Fisher, R. B. 1996. An experimental comparison of range image segmentation algorithms. *IEEE PAMI* 18(7):673–688.

Kenney, J. F., and Keeping, E. S. 1962. *Mathematics of Statistics, Pt. 1, 3rd ed.* Princeton, NJ: Van Nostrand.

Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill Companies.

Rivlin, E.; Dickinson, S.; and Rosenfeld, A. 1995. Recognition by functional parts. *CVIU* 62(2):164–176.

Stark, L., and Bowyer, K. 1994. Function-based generic recognition for multiple object categories. *CVGIP: Image Understanding* 59(1):1–21.

Sutton, M. A.; Stark, L.; and Bowyer, K. 1994. Gruff-3: Generalizing the domain of a function-based recognition system. *Pattern Recognition* 27(12):1743–1766.

Ullman, S. 1995. *High-Level Vision, Object Recognition and Visual Cognition*. The MIT Press, Cambridge, Massachusetts, London, England.

Winston, P. H.; Binford, T. O.; Katz, B.; and Lowry, M. 1983. Learning physical descriptions from functional descriptions. *AAAI-83* 433–439.

Woods, K.; Cook, D.; Hall, L.; Bowyer, K.; and Stark, L. 1995. Learning membership functions in a function-based object recognition system. *Journal of Artificial Intelligence Research* 3:177–222.