Functional 3D Object Classification Using Simulation of Embodied Agent

Ezer Bar-Aviv, Ehud Rivlin

Department of Computer Science
Israel Institute of Technology - Technion
{ezerb,ehudr}@cs.technion.ac.il

Abstract

This paper presents a cognitive-motivated approach for classification of 3D objects according to the functional paradigm. We hypothesize that classification can be achieved through simulation of actions meant to verify whether a candidate object fulfills a certain functionality. This paper presents ABSV: *Agent Based Simulated Vision*, a novel approach that tries to imitate the way humans perform certain classification tasks. ABSV can determine the category of a candidate object by verifying certain functional properties that the object should possess. Unlike conventional functional approaches, it uses virtual environment to simulate the interaction between the object and various examination agents to expose those functionalities. To demonstrate our approach we have implemented it for the recognition of several object categories. We achieved promising classification results using both complete CAD models and real 3D scanned data generated from a single view point. We believe that the concepts introduced in ABSV will influence significantly the design of robot classification systems.

1 Introduction

Object classification is considered very difficult mainly because of the huge variety in the shape of objects that belong to the same category (see Figure 1). Instead of focusing on the object's shape, the functional approach concentrates on the way a it is used or acted upon, assuming a direct link between object's functionality and its category. Traditional functional approaches often assume that the object is given as a set of labeled parts. The concept of functionality is then simplified by decomposing it into a series of rather simple geometric measurements intended to quantify the properties of each part and to analyze the relation between them [11]. For instance, classifying object as a chair requires finding two surfaces (whose dimensions and relative orientation are in specific range) that may supply seat and back support.

We believe that the connection between object's functionality and the agent that benefits from that functionality should be emphasized. We propose the ABSV (*Agent Based Simulated Vision*) approach as a way for classifying objects using simulation. We believe that in some scenarios where we are asked to classify an object, a process of simulation is

happening in our brain, whose purpose is to check for object's functionality. This claim is partly supported by cognitive psychology. According to motor-cognition we can consciously imagine or simulate actions. These actions contribute to the representation we build for the different objects we come up against.

PET studies that were done in [6] show that during perceptual analysis, in which no action occurs, we use resources that pertain to the dorsal pathway (which believed to be connected to actions). Other experiments [13] suggest that subjects perform unconscious simulation before every task, and that the response time increases with the difficulty of the task. A similar pattern of activation of action-related areas is found even in the implicit cases of observing actions or even hearing action verbs [10]. Moreover, simulation theory [12] suggests that we understand the action performed by others simply by simulating these actions. This idea was encouraged by the discovery of the mirror-neurons in monkeys [12]. Mirror-neurons respond when a monkey executes certain kinds of actions or when it perceives the same actions being performed by another monkey. The mirror-system discovered in humans is the believed to be the area that allows us to replicate action performed by others.

As mentioned, we believe that functionality can be inferred through simulation. For any functionality there should be a corresponding virtual agent. Verifying whether an object possesses certain functionality will require simulating the interaction between that object and its corresponding virtual agent. For instance, the corresponding agent of the "seatable" functionality (i.e. something we can seat on) is a virtual human model. By embodying that model in the system, we can classify objects as chairs simply by trying to make the virtual human seat on them. Note that the use of an agent encapsulates the geometric measurements done in traditional functional approaches because if the agent can seat on an object it means that the supporting surfaces are large enough, their relative orientation is suitable for human seating and so on.

2 Related Work

The problem of classification concerns the association of visual input with a category. During the years, several classification approaches have emerged. Part-based recognition approaches [2, 3], represent object categories as a set of parts in a possibly deformable configuration. Recent works used probabilistic category models and employed learning algorithms to learn the model parameters. Different works vary widely on the way parts are detected and represented and on the way learning is employed. For instance [7] models categories as flexible constellation of parts and uses a scale invariance feature-detector, while [1] represents images of the training set using a vocabulary that is automatically constructed from a set of sample images of objects that belong to the category of interest. Learning object category often requires huge training set.

Another group of works concentrates on 3D object recognition. The spectrum includes variety of works such as skeleton, medial axis representation, spherical harmonic representation and regional shape descriptors. The part based approach is exploited here as well. For instance, [9] determines the object class using learned parts classes and part to object mapping. In robotics, statistical machine learning often employs active vision techniques (or active perception) in which the robot is learning through manipulating objects [8]. Others works use learning by imitation or by observation [5].

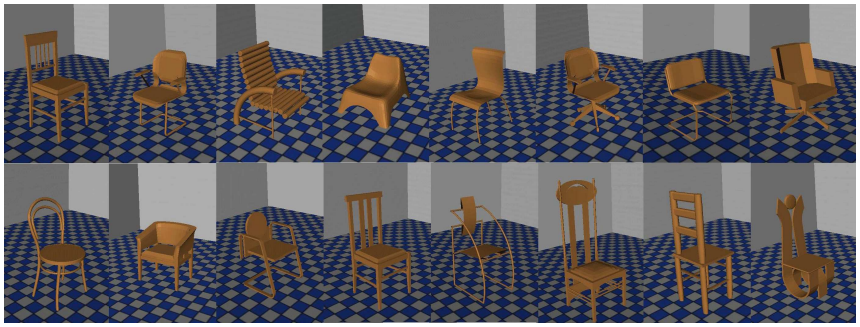

Figure 1: Part of 3D CAD DB of chairs.

According to function-based approaches, classification should refer to the functional description of the object rather then to the structural one. A representative work is [14], that presents a classification system for 3D objects under the domains of furniture, dishes and hand tools. Functional properties of each class were defined using knowledge primitives that imply about shape, for instance the height, area and relative orientation of surfaces. Some works [11] have tried to reason about the functionality of object's parts. A hummer, for instance, is recognized by a handle and a striking surface. Other works [4] have attempted to determine functionality from motion.

Our work uses the function-based paradigm, according which a category is represented by functionalities that need to be fulfilled. By embodying the receivers of these functionalities as agents within a virtual environment, we will be able to look for fully-functional configurations in which the agents satisfy the object's functionality. The use of agents encapsulate knowledge primitives and allows cognitively based simulation of actions. By imitating the way we hypothesize humans perform classification, we can directly classify objects without the need of pre-segmentation or complex shape model of categories.

3 Category Model

According to our notation, each category is characterized by n functionalities that can be either primary or secondary. Primary functionalities represent the "essence" of the category while secondary functionalities represent other desirable attributes that can be verified directly (e.g., stability). We hypothesize that primary functionality can be verified through simulation with a corresponding agent. The virtual 3D-agent serves as examination object that exposes the primary functionality of the candidate object through interaction. We refer to a category as *active-category* in case that at least one primary functionality requires the instances of the category to be active during the simulation (e.g., in the case of scissors). Otherwise, we will refer to it as *passive-category*. In that case, all primary functionalities can be verified using one static pose of the candidate object.

Our approach is specifically appropriate for man-made objects that were designed to fulfill certain functionalities. These functionalities can be revealed by embodying the receivers of each functionality as virtual agents. Current work will concern classifying such designed-objects, with obvious functionalities, that belong to passive-categories. Dur-

ing simulation, the candidate object will be static, while the different agents will search the configuration space looking for configurations that justify the corresponding primary-functionalities. The interaction will take place in a 3D virtual environment by means of collision detection.

3.1 Configuration Space

Every agent has at least six degrees of freedom (DOF's) referred as the global-DOF's. These global-DOF's pertain to the global rotation and translation of the agent in the virtual environment. In addition, an agent may have additional inner DOF's, connected to its inner structure. For each primary-functionality, we look for specific configuration justifying it. For instance, verifying that an object is "seatable" (a primary-functionality of the category chair) involves looking for configuration in which a virtual human (i.e., the agent) is seating on that object. Later, we will see how the search in the configuration space can be simplified using a cognitively based search that concern searching for semi-functional configurations.

3.2 Category Model Structure

Since every agent is unique and meant to expose certain functionality, we need to define a corresponding *environment* for every agent. Category model is represented as pair $C = \langle F, E \rangle$, where F is the set of n characterizing functionalities and E is the set of m corresponding virtual environments in which the simulation of the m primary-functionalities will occur. The secondary functionalities $\{f_{m+1}, .., f_n\}$ will be verified directly, without the need for agent simulation. Environment E_i should contain the following:
Agent - 3D model of the virtual agent A_i needed for verifying the existence of the primary functionality f_i. Embodying the agent's model encapsulates the various knowledge primitives and ensures that the object can fulfill the certain functionality.
Maximal configuration - This is the initial configuration of the agent. In a way, it expresses a cognitive-insight, by representing an initial inner configuration that can lead to almost any other inner configuration using only steepest-descent-like iterative process.
Anchor predicate - Indicates on *semi-functional* configurations, from which the iterative process mentioned above can begin. Searching for configurations that satisfy the anchor predicate allows us to perform a functional-pruning of the configuration space.
Iterative process - Takes place when the anchor predicate is satisfied. It involves a steepest-descent-like process of several inner DOFs of the agent. If it ends within a goal configuration, it implies that the specified primary-functionality is fulfilled. The category model should specify the exact iterative process (the DOFs involved, their order of activation etc).
Goal state predicate - The category-model should hold a predicate indicating goal configurations. Goal configuration is a configuration in which the specific primary-functionality is fulfilled.

4 Object Classification Framework

We are interested in classifying objects into categories. For that purpose, the system should hold a category-model for every category it needs to be familiar with. Given

an object, its primary-functionalities are revealed through interaction with appropriate agents, rather than by examining the object alone. Using a cognitively-motivated heuristic we manage to perform an efficient two-phase search for configurations that fulfill the primary functionalities. In the first phase, the agent is searching the 6-D global-DOF's configuration space, looking for semi-functional configuration. The second phase involves a steepest-descent like iterative process that iteratively changes the inner-DOF's of the agent, using a cognitively motivated heuristic, towards the goal configuration which satisfies the functionality. Both the initial state, the semi-functional configurations that we look for and the iterative process are agent-specific and should be explicitly predefined within the specific environment of the category model. They all motivated by the way humans try to fulfill that specific functionality.

Glass, for example, is characterized by three functionalities: the two primary functionalities graspable and container, and the secondary functionality stable. Classifying an object as a glass involves two simulations with two different agents, one for each primary-functionality, and a direct check for object stability (e.g., by means of calculating the center of mass). Grasping, for instance, is verified using a simulation of the interaction with virtual human palm (the corresponding agent). The maximal configuration is defined as that of the virtual palm wide open. In a sense, when starting the iterative process from that configuration we can grasp every object which is graspable. The anchor predicate indicates on a contact between the object and the center of the virtual palm, a point from which the iterative process of grasping can begin. It is done by iteratively closing the fingers (the inner DOFs) and checking for goal state (full grasping) at the end of the process.

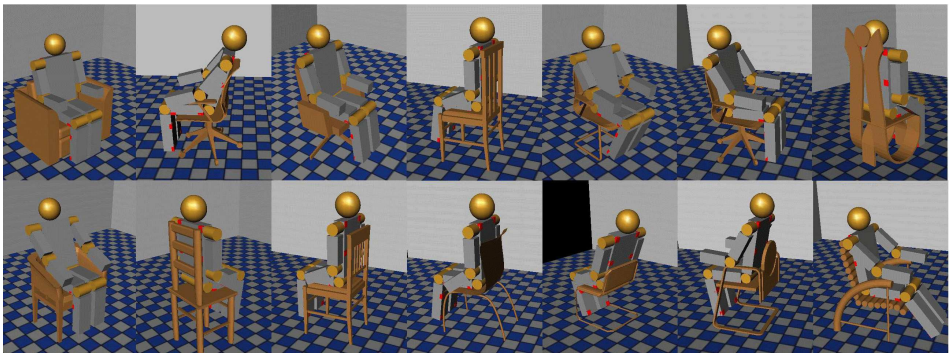

Figure 2: Chair classification using simulation of embodied virtual human agent

4.1 Meta Algorithm

We will now introduce a general algorithm for object classification. Given an object and a category, Algorithm 1 determines whether the object is an instance of that certain category. It first looks for semi-functional configurations in the global-DOF's configuration space and then performs an iterative process from each one of them towards the goal configuration.

Notice that the set of final configurations $\{final_i\}_{i=1}^{m}$ represent the configurations that best demonstrate each functionality fi. Hence, configuration $final_i$ can help the robot

Algorithm 1 : **Is_Instance_Of**(Obj, C)

for all primary functionality $f_i \in C$ **do**
  Position A_i in its maximal-configuration con_i
  $S \Leftarrow$ Search_Semi_Functional_Cons(Obj, E_i)
  $S_i \Leftarrow \{\}$
  for all $con \in S$ **do**
    $final \Leftarrow$ Iterate_Inner_DOF(con, E_i)
    if Goalp$(final)$ **then**
      $S_i \Leftarrow \langle$ Grade$(final), final\rangle$
    end if
  end for
  if $S_i = \emptyset$ **then**
    return FALSE
  end if
  $final_i \Leftarrow con$ with the highest grade in S_i
end for
return TRUE

determine the best way to position itself in order to fulfill that certain functionality. For instance, it can recommend the best way to grab an object, or the the most suitable way to seat on a chair.

4.2 Finding Semi-Functional Configurations Using Collision Detection

Searching for semi-functional configurations involves collision detection queries for each configuration in the searching grid. Algorithm 2 searches only the 6-D global-DOF's configuration space and returns the set of semi functional configurations. It performs two collision detection tests for each configuration on the grid. The first one is meant to eliminate configurations which are not collision free. The second involves ε-contact test between the agent and the object. Its purpose is to test whether some parts of the agent, that imply semi-functionality, are close enough to the object to be considered as contacting it. The ε-contact test is achieved by extending the 3D-object model by ε and checking whether it collides with the agent. Notice that the extended object could be defined only once during the classification process.

The agent model should contain indications of unique areas that contacting them implies that certain semi-functional configuration is reached. On the other hand, goal configuration implies on full-functionality and therefore in that configuration there should be contact between the object and all the predefined parts of the agent. As mentioned, both goal and semi functional configurations are verified by means of rather simple collision detection queries. Our collision detection implementation has relied on a bounding volume hierarchy (BVH) model representation using axis-aligned bounding boxes (AABB's). Specifically, we have used the SOLID library [15] for collision detection of three-dimensional objects undergoing rigid motion and deformation. SOLID allows quick update of the BVH as the model is deformed and is especially suited for collision detection of objects described in VRML, such as the ones which we use.

Algorithm 2 : **Search_Semi_Functional_Cons**(Obj, E)

```
1: S ⇐ {}
2: V ⇐ Bounding_Box(Obj)
3: CS ⇐ {6-D global-DOF configuration space within V}
4: for all con ∈ CS do
5:    if not Collision_Free(con, Obj, Agent) then
6:       CONTINUE
7:    end if
8:    if Collision_Free(con,Extend(Obj, ε), Agent) then
9:       CONTINUE
10:   end if
11:   S ⇐ con
12: end for
13: return S
```

5 Results

This section presents experimental result for classification of chairs and beds 3D CAD-models according to ABSV. The first part will present classification of objects using the embodiment of a virtual human agent. The second part will show an example for non-human agent. Working with real data scanned from single view-point will be presented in the last part.

5.1 Classifying Objects Using Virtual Human

As mentioned, every category-model should contain agent-model, maximal-configuration and iterative process description. We isolate semi-functional and goal configurations using collision-detection queries with specific parts of the model.

The maximal-configuration for the category "seatable" is similar to seating position except that the arms, legs and back are stretched forward. The agent is then translated and rotated within a bounding volume surrounding the object, looking for configurations satisfying the anchor predicate which indicates on contact between the object and the "seating areas" of the agent (marked in red). Once reaching the semi-functional configuration, the virtual-human can start an iterative process of enhancing its comfortability by leaning back and dropping down his legs and arms. Basically, almost every seating-configuration is reachable from the initial configuration, hence it is called the maximal-configuration. Notice that the iterative process is not a search within the inner-DOF's configuration space, but a rather quick and one-directional process.

The results for different chair-models are presented in Figure 2. The figure presents the configurations with the best score that the agent has found. The score is based on the functionality-level of the final configuration. We can determine a threshold from which configurations are considered to be legal. Lowering that threshold will allow the robot to improvise configurations that are semi-functional. For example, Figure 3 presents configurations with lower scores that shows the variability of the solutions. Some of these configurations are definitely legitimate improvisations of seating positions.

Figure 4 presents the results for beds classification. In that case, there were several

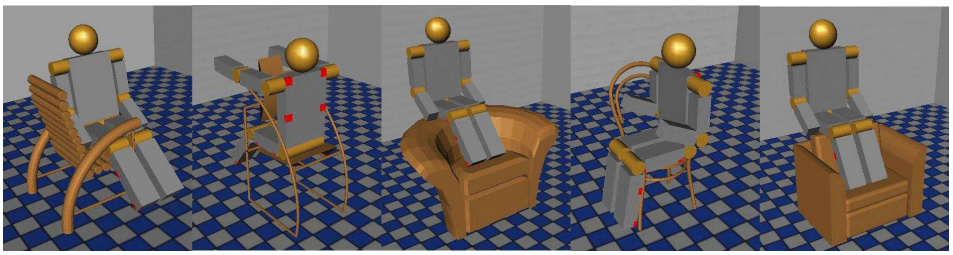

Figure 3: Some variations that got lower scores but show the variability of the solution

fully-functional configurations in which the agent is laying on the bad in different orientations. The anchor predicate for the bed category is the same as the one of chair category. The categories differ in the maximal state and the iterative process.

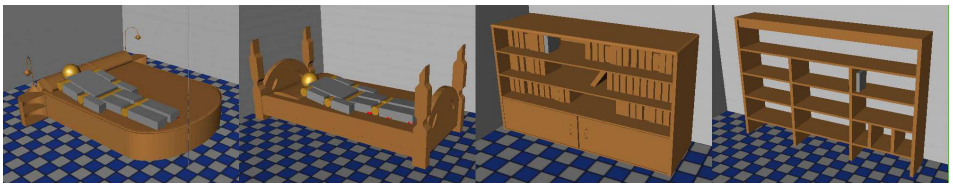

Figure 4: Beds classification using virtual human and Bookshelves classification using embodied virtual book

5.2 Classifying Using Other Agents

Although many functionalities involve virtual human agent or parts of it (i.e., virtual palm for functionality graspable), the ABSV classification approach is not confined to that agent or the functionalities driven from it. Bookshelf, for example, is a place to lay books. Therefore, a virtual book agent is required to expose the functionality of instances of that category. A candidate object is scored by the number of configurations in which the agent (i.e, the virtual book) is supported. Figure 4 presents the results of the classification. In both cases shown, we have found a place to lay down a book.

5.3 Working With Real Data

This section presents the result of classifying real-data objects. ABSV is satisfied with real 3D data coming from the robot sensors and does not require the object to be given as a set of labeled parts. The objects were scanned from a single view-point and therefore much of the object is obscured due to self occlusions. Figure 5 shows part of the real-chairs DB while Figure 6 shows the results of the scanned objects within the virtual simulation environment. The scanner has produced a 3D points cloud which were merely connected to generate a 3D triangle mesh. The use of simulations helps facing the difficulty of self occlusions and noise (i.e, the noise in the mesh, holes within the surfaces etc.). As before, the agent is searching for fully-functional configurations. In case that there exists a noticeable seating configuration from the scanning view-point, that configuration will

be found during the simulation. As can be seen from Figure 7, view points from which there is sufficient information on object's structure allow correct classification.

Figure 5: Part of the real-chairs DB

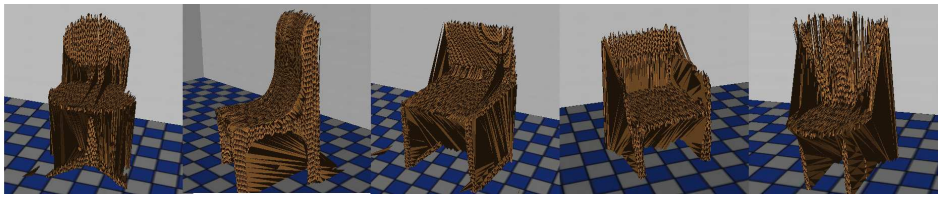

Figure 6: Part of the real-data models generated by single view-point scanning

6 Conclusion

We have presented a cognitively-motivated approach for functional 3D object classification using embodiment of virtual agents. Classification is achieved by simulating the interaction between the candidate object and several embodied agents. We have modeled each category by a set of characterizing functionalities and used simulation to look for evidence-configurations showing fulfillment of these functionalities. Each functionality had its own corresponding agent which is the receiver of the functionality. We have presented a two phase cognitively-motivated searching algorithm. Our algorithm uses functional pruning of the search space by isolating semi functional configurations, from which a one-directional steepest-descent like process can begin. Finding semi-functional and goal configurations involves only two quick collision detection queries supported by our deformed BVH-tree. We have tested the algorithm with different categories and used both CAD objects with complete model and real-data objects that were scanned from a single view-point. The algorithm managed to classify the objects correctly despite noise and self occlusions.

References

[1] S. Agarwal and D. Roth. *Learning a sparse representation for object detection*. ECCV 2002, Vol. 4, pp. 113-130.

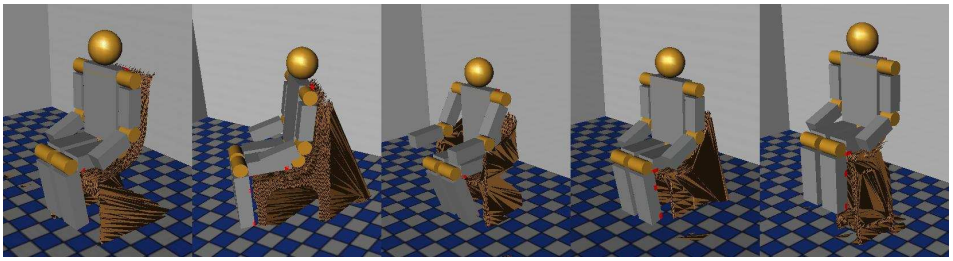

Figure 7: Classification of real-data chairs using embodiment of virtual human agent

[2] M. Burl, M. Weber, and P. Perona. *A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry*. ECCV 1996, Vol. 2, pp. 628-64.

[3] D. Crandall, P.F. Felzenszwalb, and D.P. Huttenlocher. *Spatial Priors for Part-Based Recognition Using Statistical Models*. CVPR 2005, Vol. 1, pp. 10-17.

[4] Z. Duric, J. Fayman, and E. Rivlin. *Function From Motion*. PAMI 1996, Vol. 18(6), pp. 579-591.

[5] M. Ehrenmass et al. *Observation in programming by demonstration: Training and execution environment*. In IEEE Int. Conf. on Humanoid Robots, 2003.

[6] I. Faillenot et al. *Visual pathways for object-oriented action and object recognition: Functional anatomy with PET*. Cerebral Cortex 1997, Vol. 7, pp. 77-85.

[7] R. Fergus, P. Perona, and A. Zisserman. *Object Class Recognition by Unsupervised Scale-Invariant Learning*. CVPR 2003, Vol. 2, pp. 264-271.

[8] P. Fitzpatrick. *Object lesson: discovering and learning to recognize objects*. In IEEE Int. Conf. on Humanoid Robots, 2003.

[9] D. Hubber, A. Kapuria, R. Donamukkalla, and M Hebert. *Parts-based 3D object classification*. CVPR 2004, Vol. 2, pp. 82-89.

[10] D. Perani et al. *Different neural systems for the recognition of animals and man-made tools*. NeuroReport 1995, Vol. 6, pp. 1637-1641.

[11] E. Rivlin, S. Dickinson, and A. Rosenfeld. *Recognition by functional parts*. CVIU 1995, Vol. 62(2), pp. 164-176.

[12] G. Rizzolatti et al. *Localization of grasp representations in humans by PET: 1. Observation versus execution*. Experimental Brain Research, 1996, pp. 246-252.

[13] A. Sirigu et al. *The mental representation of hand movements after parietal cortex damage*. Science 1996, Vol. 273, pp. 1564-156.

[14] L. Stark and Bowyer K. *Generic Object Recognition Using Form and Function*. World Scientific, 1996.

[15] G. Van den Bergen. *Efficient Collision Detection of Complex Deformable Models using AABB Trees*. Journal of Graphics Tools, Vol. 2(4), pp. 1-13, 1997.