2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Predicting Object Functionality Using Physical Simulations

Lauren Hinkle and Edwin Olson

Abstract— **It is challenging for a robot acting in the world to interact with and use novel objects. While a person may be able to look past visual differences and recognize the intended function of an object, doing so is more difficult for robots, which tend to rely on visual similarity to recognize categories of objects. A robot that recognizes and classifies objects based on their functional properties and potential capabilities is better prepared to use unknown objects.**

We propose a technique for functionally classifying objects using features obtained through physical simulations. The described method simulates spheres falling onto an object from above. We show how a feature vector can be derived from the results of the physics-based simulation, and that this feature vector is informative for a variety of affordance classification tasks. This process allows a robot equipped with a 3D sensor to determine the functionality of objects in its environment given only a few training examples from various function classes. We show that this method is able to accurately learn membership of 3D models in three function classes: "drinking vessel", "table", and "sittable". We then show that this can be extended to 3D scans of objects using the models as training examples.

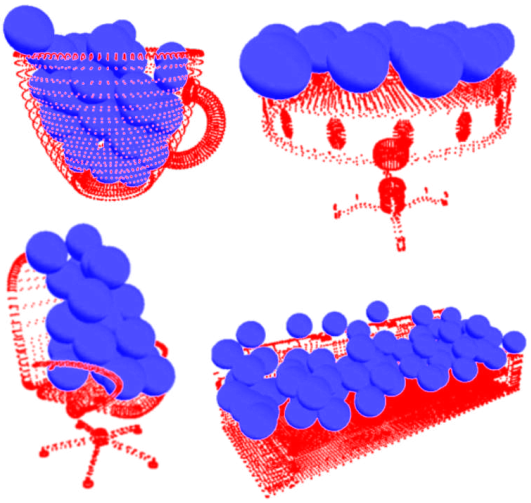


Fig. 1: The result of simulating spheres falling onto a cup, table, chair, and sofa (not shown to scale).

I. Introduction

The dominant approach in object classification is to use the appearance of objects: a chair might be recognized because particular visual features (e.g. SIFT features [1]) are associated with it. The challenge with this approach is that visual appearance within many classes, such as chairs, is highly diverse, making learning difficult. What all chairs have in common, however, is that they serve a specific physical function: their construction affords the specific interaction of "sitting". This shared functional property, or affordance, is what allows all chair-like objects to be grouped in a class. Functional properties can be used to classify objects and to give insight into how an object can be used and by whom.

In this paper, we approach the problem of object classification by considering the physical properties of objects. Our hypothesis is that these physical properties, which we predict by computing a 3D model of the object and subjecting it to a series of simulations using a simple physics engine, will be highly predictive in object classification tasks. We test this hypothesis using a limited set of physics-based simulations and show how machine learning features can be extracted from the results.

Functions and affordances are intrinsically linked to the physical properties of an object, but the exact mapping between them is unclear. Previous work approaches function recognition and classification in a variety of ways including predefining characteristics of function classes [2], [3], analyzing images or video of people interacting with objects [4], [5], [6], and, more commonly, having a robot learn functional properties by either interacting with objects directly [7], [8] or in simulation [9], [10]. These techniques require a large time commitment and focus only on the functionality afforded to a given user, either the human being watched or the robot that is interacting. Learning the actions a given robot can perform with an object misses many of the possible functions and uses of the object. These more general properties can't be discovered through an individual's interactions, but can be explored in simulation.

We introduce features for predicting object functionality obtained by physical simulation. In particular, we simulate the effect of dropping spheres onto an object from above, allowing them to roll and settle on the object. The resulting distribution of spheres is analyzed and used for function classification. We show that objects in the same function class have correspondingly similar distributions of equally sized simulated spheres.

In this paper we:

- Propose a class of object features based on physical properties determined through simulation
- Experimentally show the proposed framework can support learning functional classes like "sittable", "table-like", and "drinking vessel" using 3D object models
- Demonstrate successful function classification on examples of real-world objects using sensory data from a Kinect.

The authors are with the Computer Science and Engineering Department, University of Michigan, Ann Arbor, MI 48104, USA {lhinkle,ebolson}@umich.edu http://april.eecs.umich.edu

II. Prior Work

Prior work on function classification can largely be divided into systems that reason about objects' physical properties and systems that learn functionality through interaction [11], [12].

A. Reasoning About Functionality

The initial work on distinguishing function classes emphasized measurable quantities of objects, such as their dimensions, with pre-defined parametric requirements for each class. Among the first function classes explored was "sittable." Stark et. al use a list of the faces and vertices defining an object to determine whether it can be appropriately used as a chair [3]. The list helps determine characteristics such as the object's dimensions, relationships between its faces, and its predicted center of gravity. These higher-level features are compared to a parameterized structural model defining a sittable object. The model isn't learned, but rather is specified with hard-coded "knowledge primitives" that define which relationships must exist.

Rivlin et al. introduced a method to determine the overarching functional properties of an object by recognizing and relating the function of parts of the object [2]. By combining shape primitives such as the dimensions, spatial primitives describing the relationship between parts, and pre-defined functional primitives, they are able to recognize hammers in a scene.

Rather than using size and shape directly, Horton et al. use sensory feedback from an AIBO to discover visual symmetries and what they term "inverse symmetries". They use these symmetries to predict relationships between objects, allowing an AIBO to use a tool to push an object over rough terrain [13].

B. Learning Functionality Through Interaction

The majority of work on recognizing the functional properties of objects is framed in terms of affordances. The term "affordance" comes from ecological psychology [14], but has been adopted and adapted by other communities including human computer interaction and artificial intelligence. An affordance is often interpreted as a perceivable attribute of an object that alerts the perceiver as to what functions he, she, or it can performed on or with the object. Approaching function recognition through the lens of affordances has led to an emphasis on either analyzing images of people interacting with objects or having robots interact (or simulate interacting) with objects.

Systems that analyze how humans interact with objects attempt to recognize where people grasp objects and the motions they perform with them. This is often done by analyzing a series of frames to determine when and where contact is made and tracking the ensuing motion [4], [5], [6].

Most methods that use robot interaction employ a "babbling" stage, in which the robot interacts freely in the environment without any goals, to determine the functions it can perform on nearby objects. Stoychev created a system that learns to use sticks with a variety of end-effectors to move a puck around a table [8]. After hundreds of test trials, the robot learns to associate the puck's trajectories with the color-coding of the stick that caused them and is able to choose the correct stick to move the puck from a start position to a goal [10]. Similarly, Brown and Sammut created a simulated robot that employs babbling to learn to use a stick to push balls out of tunnels and to climb ramps to reach rewards. The robot learns numerical relationships between the tools it uses and the tasks they help perform (for example, that the stick must be thin enough to fit in the tunnel) [9].

Although tool-use is the most common emphasis in affordance and function recognition, other affordances such as traversibility are also explored through robot interaction and babbling. Erdemir et al. explore traversibility using statistical information obtained in a babbling stage to perform internal rehearsals before acting in later stages [7]. Uğur et al. consider traversibility of a robot among objects that can or cannot be moved, such as balls or boxes respectively [15]. In a departure from both tool-use and traversibility, Griffith et al. use vision and sound to determine whether objects are containers. A robot accomplishes this both by attempting to manipulate a small toy inside the object [16], and by moving objects under running water in a sink [17].

These affordance-driven techniques allow the robot to learn what actions it is capable of performing with an object, but do not reveal wider, functional properties of an object. For example, although an object might provide an excellent seat for a person, most robots are incapable of sitting and will therefore never discover this affordance. Simulation provides an arena for exploring properties a robot cannot discover through its own interactions but which may be useful to recognize when communicating with humans or other robots with differing capabilities.

Bar Aviv and Rivlin use simulated "examination agents" that interact with objects to determine the objects' functional properties [18]. Each function category is defined as a combination of an examination agent and a set of constraints on the location and orientation of the agent, both of which must be pre-defined. For example, in order to test if an object can be sat upon, a human model is manipulated on the object and tested for "sitting positions", parameters in which the angles of the model's joints must lie. If the model can be positioned within these constraints, the object is functionally categorized as a chair. They are similarly able to recognize tables using a simulated person and bookshelves using a simulated book.

The method described in this paper also explores the use of simulation to predict the functional properties of objects without restricting functions to those performable by a particular robot or person. However, unlike Bar-Aviv and Rivlin's method, function classes do not have predefined "examination agents" or goal orientations.

III. Simulated Interaction for Function Classification

Our method is, in short, to simulate dropping spheres onto an object and use the resulting distribution to functionally classify it. We use a 3D point cloud of an object to predict how spheres dropped on the object from above will respond to it. The simulated spheres follow basic physical rules when colliding with either the object's model or other spheres before coming to rest in static locations. Each function class is associated with a sphere radius that provides the most informative distribution, i.e. that allowing for the best classification accuracy, which is learned using initial training examples. We derive a feature vector from a histogram of the locations of the spheres on the object.

We define the functional properties of an object to be how the object is able to interact with other objects, specifically, how it affects them and how it is affected by them. Coarse simulation of these interactions predicts how an object might react in the real world without requiring any actual interaction to occur. The benefits of this are that it's often faster than initiating physical interaction and that one can explore interactions that are irreversible in the real world. Additionally, it may allow for the discovery of a more general set of functions and affordances. We hypothesize that even if the simulated interactions don't perfectly model the real-world interactions they are designed to imitate, they reveal useful information about the object. We choose spheres for our preliminary interaction simulation tests because their physics are straightforward and they are able to roll and fit the shape of objects.

A. Simulating an Object and Falling Spheres

Although an object mesh provides more information and results in more accurate simulations, we use only a point cloud for simulation to reflect the data obtained from 3D sensors. A robot exploring the world with a 3D sensor does not have the advantages afforded by an object mesh. While it can easily build a point cloud of its environment, transforming the point cloud into a set of faces is a computationally expensive and unnecessary task. Basic spatial knowledge of the point clouds is assumed by creating a ground plane along the lowest Z-plane of the bounding box. Spheres fall toward this plane, in a simulation of gravitational effects.

A more sophisticated physics simulation system like Open Dynamics Engine [19] could be used for our approach. However, high-fidelity results are not necessary for good classification performance. Consequently, we use a low-fidelity simulation system that runs much more quickly. In our system, simulated spheres drop one at a time from randomly chosen (x, y) positions above the object's point cloud. A sphere drops straight down until it collides with points in the point cloud or another sphere. The sphere "rolls off" the points it collides with by estimating a plane through the points and rolling along the plane. Similarly, a falling sphere rolls off spheres that have already settled. A sphere continues to roll along the object until it either collides with an object that blocks its path, comes to rest on a flat surface, or rolls off the object entirely.

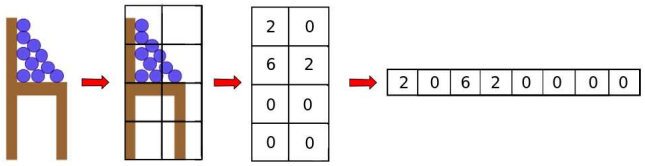


Fig. 2: A simple 2D example of how features are extracted. Once the simulation is complete, the object is divided into equal bins. The number of sphere centers in each bin is counted and these counts form the histogram and feature vector.

Once a sphere comes to rest, it is static and does not move regardless of other spheres colliding with it. The histogram is calculated using only spheres that land on the object; spheres that roll off the object and strike the ground are ignored.

Spheres continue to fall onto the object until a user-defined number of sequential spheres either roll off or fail to be placed on the object. Together, these two conditions suggest that the object is covered by spheres and cannot hold more. Although this threshold could be varied for different object sizes, in our tests, we set this threshold to 50 for all objects.

Spheres pile upon one another without direct support from the object while the pile fits within the object's bounding box. This compensates for the fact that spheres become static once they have landed. In the real world, a ball falling on top of or rolling up against another will generally cause the second ball to roll as well if it is not held in place; a pile of balls doesn't form without a structure to support them. The result of this constraint is that cups fill up with spheres, chairs have a slope growing from the front edge of the seat to the back, and tables have a flat surface of spheres. This can be seen in Fig. 1.

B. Feature Extraction

The final location of the spheres is summarized by a histogram, where each element of the histogram represents the number of spheres that fell within a given box in space. A simplified, two-dimensional example of the histogram and features resulting from spheres being dropped on a chair is shown in Fig. 2.

Histograms are chosen to represent the results of the physical simulation because we predict different objects will cause the spheres to land and group in different ways. The distribution of the spheres is affected by their radii as well as the size and shape of the object. Prior work uses object size and shape as features for recognizing function classes, and although the proposed method does not use these properties explicitly, the resulting vectors are affected by them. The distribution of the spheres is summarized using a histogram that divides the object into equally sized cubes.

The initial feature vector for each object is constructed with the histograms derived from a variety of sphere sizes. It is reasonable to believe that different sphere sizes will perform better for different function types, which may vary

greatly in expected size themselves. For example, large spheres may not be helpful for recognizing drinking vessels because they can support only one large sphere, while such spheres may be more relevant in determining objects that are sittable for a person. To explore this, a variety of radii are initially used. In practice, we have found that smaller spheres, on the scale of a few centimeters, lead to better classification for all function types because they provide richer histograms.

The accuracy of the histograms depends upon the discretization chosen. Large bins will encapsulate whole objects, causing spatial relationships between spheres to be lost. Alternatively, bins that are too small will only contain a single sphere center, making comparisons between feature vectors of different objects harder because spheres are required to land in the same location rather than in the same general area. After testing several alternatives, including a range of fixed sizes of bins and sphere radius-to-bin ratios, we found that bins three to four times the diameter of the spheres perform best. This means the histograms resulting from different sphere radii have different numbers of bins and are not comparable. Additionally, this results in a single histogram bins of some of the larger spheres fully containing some of the smaller objects. This is non-ideal, but was not further explored in this work.

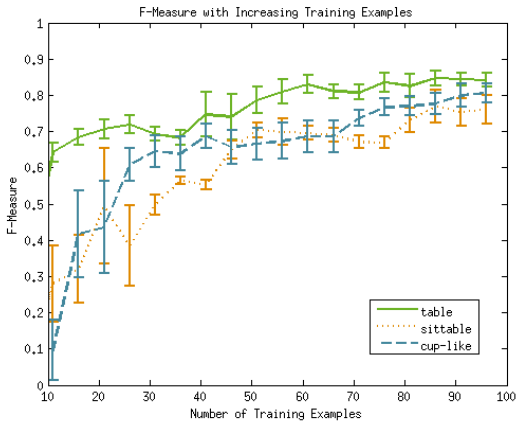
C. Function Classification

Each function class is associated with a single radius that is found to produce sphere distributions that result in accurate classifications. In order to determine the best-performing radius for a given function class, initial examples of each class are divided $70\% - 30\%$ into a training and a validation set. Function classification is performed using a binary, one-vs-all, classifier for each class. In our evaluation we use a weighted nearest neighbor classifier where training examples in the classifier are weighted with the inverse Euclidean distance between an example and the histogram being evaluated. A weighted nearest neighbor classifier was chosen because we expect to have few initial training examples, and we have found the weighted nearest neighbor classifier gives good predictions with few training examples. Individual binary, one-vs-all classifiers are necessary because they do not restrict function class membership to a single class. Thus a sofa, which can be either sat or laid upon will have positive classification for "sittable" and "layable" classifiers but will be negative for classifiers such as "table".

The F-measure is chosen as our evaluation metric because the goal of function classification is to correctly identify as many objects exemplifying the function as possible while minimizing the number of false positives. The F-measure accomplishes this by heavily penalizing either low precision or low recall.

$$\text{F-measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Each radius is evaluated by calculating the F-measure of its classification for each function class on the validation set. The radius with the highest F-measure for a given function is chosen as the representative for that function class and is used in the future when determining whether an object is a member of that class.



(a) The average F-measure obtained with an increasing number of total training examples over 10 trials. Error bars denote variance.

Class	Radius (cm)	F-Measure	Accuracy
Cup-like	0.5	0.80	0.96
Table-like	1	0.85	0.94
Sittable	0.5	0.74	0.92

(b) Ultimate radius chosen for each function class and the F-measure and accuracy at that radius.

Fig. 3: Performance of function classification using features derived from simulating falling spheres.

When functionally classifying an object, only the chosen radius for a given class is tested on the object. A robot looking for sittable objects in its environment would only simulate dropping spheres of a specific radius to make a classification. Alternatively, if the robot wanted to determine which function classes an object exhibited, it would simulate dropping spheres with the representative radius for each function class and consider the resulting distributions individually.

This radii selection test can be performed either offline with 3D object models or online using the first several examples of the functional class the robot sees while exploring the world.

IV. Evaluation

We evaluate how well our method is able to functionally classify three classes, "table", "drinking vessel", and "sittable", by testing the classification accuracy for a collection of 3D models. We are able to achieve high accuracy and recall in identifying objects. We further evaluate function recognition with real-world data from a Kinect, using the 3D models as training examples.

A. 3D Model Dataset

For testing and evaluation, we created our own database of 200 CAD object models that were individually downloaded

	cup-like	table-like	sittable	none
cup-like	10	0	0	6
table-like	0	20	0	2
sittable	1	0	6	15
none	2	8	0	33

(a) 20 Training Examples

	cup-like	table-like	sittable	none
cup-like	11	0	0	5
table-like	0	18	0	4
sittable	1	3	13	6
none	1	2	3	34

(b) 45 Training Examples

	cup-like	table-like	sittable	none
cup-like	12	0	0	4
table-like	0	21	0	1
sittable	1	2	18	1
none	2	3	6	29

(c) 85 Training Examples

Fig. 4: Confusion matrices showing the function classification for the same test set given differing numbers of training examples. Rows indicate actual membership while columns denote classified memberships. As more training examples are seen, function classification improves.

from several free online databases[1]. These models were downloaded as .stl or .3ds files and were converted into unordered point clouds by sampling at least one point every half centimeter along the faces. Models were rotated so that all objects have the XY plane as ground with the positive Z-axis pointing up, and were translated to the origin. Models with incorrect scales were manually adjusted to reasonable values.

Of the 200 objects acquired, 40 were a type of sittable object (chairs, sofas, benches, etc), 40 were table-like objects (dining room tables, desks, etc), and 30 were drinking vessels (glasses, bowls, etc). The others were household objects ranging from furniture such as baths, ovens, and bookshelves, to appliances such as televisions, vacuums, microwaves, and computers to other knick-knacks such as books, shoes, and lamps.

B. Recognition on Model Dataset

We evaluate how many training examples are necessary for our system to accurately classify these functional categories. Of the three function classes, we expect sittable objects to be the most challenging because the objects in it vary the most. For example, drinking vessels tend to be small and similarly sized, and most tables and desks are of similar height, although their widths and lengths vary. Sittable objects, on the other hand, range from desk chairs to sofas to bar stools, all of which have very different sizes and expected distributions.

[1] `http://grabcad.com` and `http://archive3D.net`

For each test, the dataset was split randomly into equally sized training and testing sets. As described in Sec. III-C, roughly 30% of the training set is used for cross validation to select the best radii. This makes classification challenging until a sufficient number of examples of each function type have been seen to ensure a positive example appears in both the training and the cross validation set. The radius with the highest F-measure on the cross validation set is used to evaluate the test set. The results can be seen in Fig. 3, where the total number of training examples of all classes is plotted against the F-measure for each function class. These results are an average obtained over 10 trials which used the same 200 objects but varied the test set.

The score is a result of higher recall and lower precision, as is indicated in the confusion matrices in Fig. 4, where there are more false positives than false negatives. This means more objects are identified as having a given function than actually do, but most objects that have it are correctly identified. The confusion matrices show the membership classifications for the test data given 20, 45, and 85 total training examples, and are from one of the trials averaged in Fig. 3. The rows represent the function classes the objects actually belong to, and the columns represent which functions they are classified as. Although this dataset has no overlapping function classes, membership is not restricted to a single class. Despite not having this restriction, most objects were predicted to belong to only one class. The few exceptions are evident in the confusion matrices which sum to more than 100, as this indicates a single object was classified positively by more than one function classifier.

Fig. 5 shows a matrix of function classification for each object in our dataset. Each object was functionally classified using the other 199 objects as training examples. Each row of the matrix represents predicted membership in a given function class. Additionally, "no predicted memberships" is indicated in the bottom row. In order for an object to be classified as having no functions it must be negatively classified by each function classifier, as there is no "none" classifier. Every column represents a different object from the dataset, grouped by their labeled functions. These labels are somewhat subjective as, for example, people sometimes sit on tables or desks. For this work we chose to label objects only with their primary functional property. Among the challenging objects to classify were bar stools, which have the height of a table and do not have backs like most chairs, mid-sized filing cabinets, bookshelves, and counters which were confused for tables (and which are, indeed, large flat surfaces often put to similar uses as tables), and vases and shoes which were often confused as drinking vessels.

These results show the proposed feature set can functionally classify objects that belong to a given class, though increased precision may be desired. The results were obtained using only the features derived from the physical simulation. One could imagine integrating other feature types, such as those based on appearance, into the classifier. Such features are largely orthogonal to our current feature set and would provide very different information. We hypothesize that a

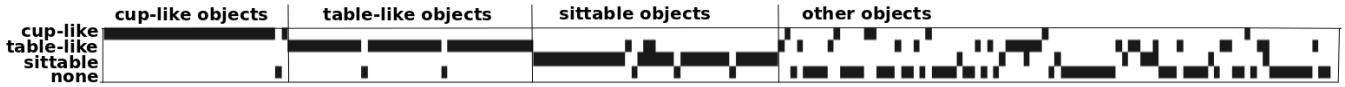
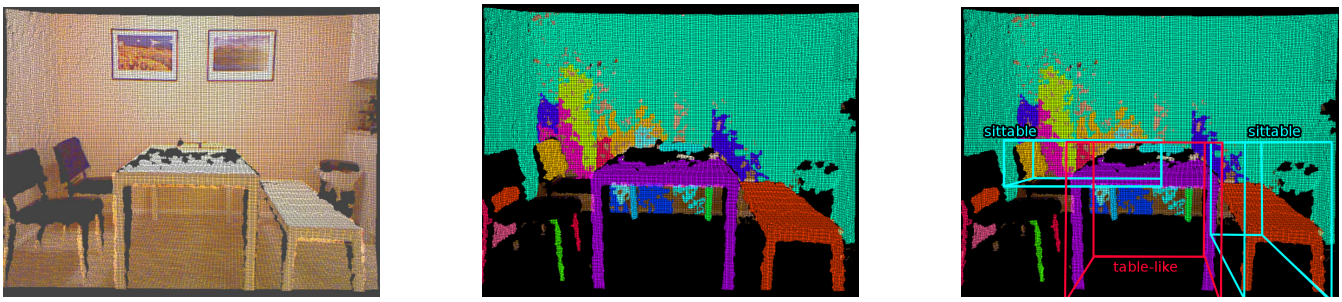


Fig. 5: Classification of each object in the 3D model dataset. This matrix shows the function classification for each object in our dataset, given the other 199 objects. If all objects were classified correctly, one would expect to see four unbroken lines. Note that there is no "none" classifier, so correctly identifying an "other" object requires *all* the other classifiers correctly classifying it as a negative example.



(a) 3D point cloud of a dining room.

(b) Segmentation of dining room

(c) Objects identified with functional properties.

Fig. 6: Object function classification on real world data. Using the point cloud of this room obtained by a Kinect, the table, bench, and one of the chairs were correctly identified with their functional properties. Objects identified with functional properties have color-coded boxes drawn around them. Cyan boxes indicate objects that are sittable, while magenta indicates table-like objects.

system that used both types of features would perform even better.

C. Recognition with Real-World Data

We evaluated our method on real data, obtained with a Microsoft Kinect, in order to investigate the performance impact of imperfect models arising from occlusion and noise. We captured several scenes that contained sittable and table-like objects, including the dining room shown in Fig. 6 as well as several office environments. Drinking vessels were not tested because their bottoms would not be observed by the Kinect and so their simulated counterpart would not be able to hold any spheres. An assumption that all objects have a flat bottom even when the bottom cannot be seen by the robot may help solve this.

The dining room shown in Fig. 6 is representative of how well functional properties were recognized in all the analyzed scenes in terms of how well objects are classified and some incorrect segmentations. Although an RGB-D scene is shown in the first image, only the depth data is used for segmentation and classification. 3D segmentation is a challenging problem on its own, and in this work we assume a correctly segmented scene. For this experiment we use scenes with few overlapping objects to maximize our simple segmentation algorithm's likelihood of segmenting correctly. A scene is segmented by identifying and removing the floor plane using RANSAC [20]. Points are then agglomerated into objects based on the Euclidean distance between them. Points within 2 cm of one another are considered part of the same object. This somewhat large threshold is a result of the Kinect's noisy range data, which varies from a few millimeters up close to 4 cm at the edge of its range [21]. We believe better segmentation algorithms should have at least equivalent function classification performance.

The point cloud for each segmented object is tested for membership in the three function categories. The full CAD model dataset provides training examples for each function type and is used to select the radius for each function class. Simulated spheres are dropped onto the point clouds using the method described. The majority of the segmented objects in the scenes (which includes several slices of the wall and the legs of the front-most chair in Fig. 6) were not positively categorized by any of the function classifiers. Objects that were positively classified are indicated with a box. The dining room table was correctly classified as a table, and the bench as sittable. The closest chair was too reflective for the Kinect to obtain depth measurements, but the further chair, while poorly segmented, was correctly identified as a sittable surface.

V. Conclusion

This work is a step toward incorporating features discovered through physical simulation into object function recognition. Additional simulated interactions, such as "dropping" balls from all directions instead of just from above, or considering multiple objects in conjunction with each other might create features with strong predictive capabilities. Such features could supplement other methods of function and affordance recognition.

In this work we explored the features that can be obtained for function classification and recognition from basic sensory data beyond the more commonly used camera and depth data. We show that informative features can be discovered by simulating physical interactions with an object. We have begun exploring this concept by simulating bombarding an object with spheres of differing radii and calculating the distribution of the spheres that land on it. Using the distribution of spheres as a feature vector, a classifier is able to predict whether the object belongs to three different function classes. We are able to learn the appropriate radius size for each function classes that leads to high classification accuracy. We obtain reasonable classifications using a dataset of 3D models, and further show real-world objects detected with a Kinect can be correctly functionally classified using the models as training examples.

ACKNOWLEDGMENTS

This research was supported by ONR grant # N00014-13-1-0217.

REFERENCES

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on Computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[2] E. Rivlin, S. J. Dickinson, and A. Rosenfeld, "Recognition by functional parts," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1994, pp. 267–274.

[3] L. Stark and K. Bowyer, "Generic recognition through qualitative reasoning about 3-d shape and object function," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1991, pp. 251–256.

[4] Z. Duric, J. A. Fayman, and E. Rivlin, "Function from motion," *Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 579–591, 1996.

[5] A. Gupta and L. S. Davis, "Objects in action: An approach for combining action understanding and object perception," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[6] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele, "Functional object class detection based on learned affordance cues," *Computer Vision Systems*, pp. 435–444, 2008.

[7] E. Erdemir, C. B. Frankel, K. Kawamura, S. M. Gordon, S. Thornton, and B. Ulutas, "Towards a cognitive robot that uses internal rehearsal to learn affordance relations," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2016–2021.

[8] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 3060–3065.

[9] S. Brown and C. Sammut, "An architecture for tool use and learning in robots," in *Australian Conference on Robotics and Automation*, 2007.

[10] J. Sinapov and A. Stoytchev, "Detecting the functional similarities between tools using a hierarchical representation of outcomes," in *Proceedings of the IEEE International Conference on Development and Learning*. IEEE, 2008, pp. 91–96.

[11] E. Biçici and R. S. Amant, "Reasoning about the functionality of tools and physical artifacts," Department of Computer Science, North Carolina State University, Tech. Rep. 22, 2003.

[12] T. E. Horton, A. Chakraborty, and R. St. Amant, "Affordances for robots: a brief survey," *AVANT. Pismo Awangardy Filozoficzno-Naukowej*, no. 2, pp. 70–84, 2012.

[13] T. E. Horton, L. Williams, W. Mu, and R. S. Amant, "Visual affordances and symmetries in canis habilis: A progress report," in *AAAI Fall Symposium Technical Report*, 2008.

[14] J. Gibson, "The ecological approach to visual perception," 1979.

[15] E. Uğur and E. Şahin, "Traversability: A case study for learning and perceiving affordances in robots," *Adaptive Behavior*, vol. 18, no. 3-4, pp. 258–284, 2010.

[16] S. Griffith, J. Sinapov, V. Sukhoy, and A. Stoytchev, "How to separate containers from non-containers? a behavior-grounded approach to acoustic object categorization," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1852–1859.

[17] S. Griffith, V. Sukhoy, T. Wegter, and A. Stoytchev, "Object categorization in the sink: Learning behavior–grounded object categories with water," in *Proceedings of the ICRA Workshop on Semantic Perception, Mapping and Exploration*, 2012.

[18] E. Bar-Aviv and E. Rivlin, "Functional 3d object classification using simulation of embodied agent," in *Proceedings of the British Machine Vision Conference*, 2008, pp. 32–1.

[19] R. Smith, "Open dynamics engine," accessed July, 2013. [Online]. Available: http://www.ode.org/

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] K. Khoshelham, "Accuracy analysis of kinect depth data," in *ISPRS workshop on laser scanning*, vol. 38, 2011, p. 1.