3DNet: Large-Scale Object Class Recognition from CAD Models

Walter Wohlkinger and Aitor Aldoma and Radu B. Rusu and Markus Vincze

***Abstract*— 3D object and object class recognition gained momentum with the arrival of low-cost RGB-D sensors and enables robotics tasks not feasible years ago. Scaling object class recognition to hundreds of classes still requires extensive time and many objects for learning. To overcome the training issue, we introduce a methodology for learning 3D descriptors from synthetic CAD-models and classification of never-before-seen objects at the first glance, where classification rates and speed are suited for robotics tasks. We provide this in 3DNet (3d-net.org), a free resource for object class recognition and 6DOF pose estimation from point cloud data. 3DNet provides a large-scale hierarchical CAD-model databases with increasing numbers of classes and difficulty with 10, 50, 100 and 200 object classes together with evaluation datasets that contain thousands of scenes captured with a RGB-D sensor. 3DNet further provides an open-source framework based on the Point Cloud Library (PCL) for testing new descriptors and benchmarking of state-of-the-art descriptors together with pose estimation procedures to enable robotics tasks such as search and grasping.**

I. INTRODUCTION

Central tasks for robots are to find, grasp and manipulate objects. While an industrial robot helper needs to know about the specific objects in production, home robots should know about all the object classes typically found in human living space. And certainly, the user expects that the robot can learn novel objects and object classes.

Especially the domestic setting with its plethora of categories and their intraclass variety demands great generalization skills from a service robot. The categories are characterized mostly by their shape ranging from low intraclass diversification as in the case of fruits and simple objects like bottles up to high intraclass variety of classes such as liquid containers, furniture, and especially toys. With robots starting to tackle real-word scenarios, we require fast and reliable object and object class recognition. Especially in robotics manipulation, where object recognition and object classification have to work from all possible viewpoints of an object, data collection for training becomes a bottleneck. Especially for classes with high intraclass variability it is required to obtain a very large number of objects in the training phase.

With the arrival of an affordable RGB-D sensor, the Kinect, and the increasing number of mobile manipulators, e.g., WillowGarage's PR2, learning classes and objects for each one of the objects and robots seems like a waste of resources. The goal should be to have a common knowledge database shared between the robots. So when one robot in place A is trained on novel objects or classes, another robot at place B can update the reference database and detect the new object classes (maybe with the exception if objects differing too greatly from country to country). This also holds true for the introduction of new features and descriptors: once introduced and integrated, everyone should be able to use these algorithms. This is especially true for researchers not working in the field of object and object class recognition, as for them, classification is a necessary step to achieve their own research to provide the robot with new functionalities.

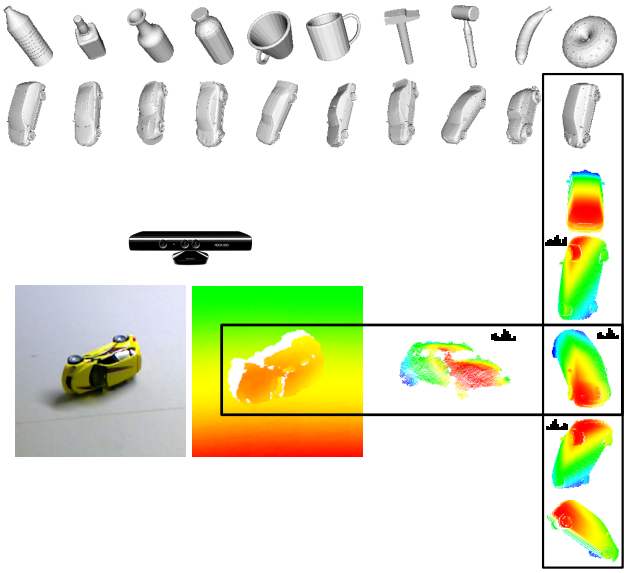


Fig. 1. System overview: For classification, the RGB-D image is segmented to obtain a point cloud cluster and to calculate a cluster descriptor. The descriptor is then compared to synthetically rendered views of CAD-models. The most similar view delivers the best 3D model, view and class label. 3DNet provides the means for this system: A large set of organized 3D CAD models and an extensible framework with needed algorithms and descriptors.

To gain momentum towards that vision, we introduce 3DNet (3d-net.org), a free resource providing training data in the form of 3D CAD models, a framework for implementing and evaluating existing and new 3D shape descriptors and out-of-the-box object recognition and object classification. Figure 1 depicts the proposed system with 3D CAD models or scanned 3D models as input for training and a view based matching against the sensor data. We encourage the community to exploit and add to this open framework, which presents state-of-the-art performance compared to

This work was conducted within the EU Cognitive Systems project GRASP (FP7-215821) funded by the European Commission.

Wohlkinger, Aldoma and Vincze are with Vision4Robotics Group, Automation and Control Institute, Vienna University of Technology, Austria [ww,aa,mv] @ acin.tuwien.ac.at

Rusu is with Willow Garage, Menlo Park, USA rusu @ willowgarage.com

Lai [11], as robotics research focused on object and object class recognition based on 3D shape from depth sensors are almost at the very beginning and the features and descriptor possibilities are not fully exploited yet. Our contributions encompass three distinct but related areas:

- First, we propose to use synthetic CAD models collected from the web. The models are organized according to WordNet [6] and are provided through 3DNet (3d-net.org). This enables the robotic community to fast and easily train many object and object class recognition algorithms without tedious object scanning.
- Secondly, we provide an open-source framework based on PCL [18] with state of the art descriptors for use with the 3D model database. These descriptors are automatically trained on the 3d models and enable real-time, high performance object and object class recognition to be integrated into common robotics frameworks such as ROS [15]. The framework provides templates to easily integrate new descriptors.
- And third, we propose benchmarks with increasing complexity to be used as test environment to enable objective comparison of descriptor performance. The benchmark datasets are in addition to the RGB-D dataset by Lai [12] and the SHREC Range Image Retrieval Contest [5] where we provide out-of-the-box evaluation scripts to be tested on these already available datasets. To provide an unbiased test dataset for our 200 category model database, datasets from the community are being added via 3DNet.

After reviewing related work we present the 3DNet database in in Section III and the classification framework in Section IV and present benchmarks and evaluation results in Section V.

II. State Of the Art

Our reference is the hierarchical RGB-D object test dataset that was made available to the community by Lai [11]. It presents 51 object classes also organised according to WordNet relations. In the accompanied approach [12] multiple features are combined, trained, and evaluated on this dataset and the authors showed that shape together with color leads to improved object recognition. Although the authors collected a large dataset from multiple viewpoints, the authors did not make the code nor evaluation tools available to the community. The KIT Object Models Web Database[1] is also a free resource of 3D models with texture scanned with a structured light setup representing mostly household items. The closest benchmark to our system is the SHREC Shape Retrieval Contest of Range Scans[2] where a set of 800 3D models in 40 classes is given as target set and 120 range scans captured with a Minolta Laser Scanner and converted to meshes are given as query set. The results were presented in [5] where the top performer reached a nearest neighbor classification rate of 67.5% with a bag of words approach with depth-sift features [7].

Regarding the development of datasets, an interesting issue was brought up by Torralba and Efros [21]: Datasets (e.g., Caltech-101 or the Pascal VOC) for measuring and comparing competing algorithms are biased. This also halts true for the RGB-D dataset of Lai [11], which has a selected set of objects, poses, lighting conditions and objects on a small turntable. The authors of [21] provide suggestions to minimize the bias in datasets which include:

- Selection Bias: to avoid a bias towards human-selected images, data should be collected automatically from multiple sources, using multiple search engines from multiple countries, or use a large set of unannotated images and label them by crowd-sourcing as done with ImageNet [4] or LabelMe [16].
- Capture Bias: as objects almost always appear in the center of the image and objects tend to have a standard position (mugs upright with handle to the right). In a robotic-centered RGB-D context, the capture bias could be resolved by capturing failed manipulation attempts which lead to objects in random pose and distance to the camera and thus avoiding human-biased viewpoints (e.g., looking down 45 degrees).
- Negative Set Bias: is reduced if we add scenes to the database that do not contain any of the database objects.

These suggestions motivated us to create a publicly available community-built test dataset for the unbiased, objective and extensible comparison of classification and recognition algorithms for robotics.

III. DATABASE

The intention is to build up and maintain a steadily growing database of object classes for robotic applications. We propose to adopt the paradigm of learning models of classes from the web to easily capture intra-class variability and simplify data gathering. We also link the classes to actual scenes with (new) samples of these object classes.

To start with, we provide four CAD-model databases with increasing size and complexity accompanied with corresponding test databases. The model databases are constructed by semi-automatically downloading models from Google's 3D Warehouse and various smaller, free online repositories for CAD models[3]. The models are linked to the WordNet [6] structure, which provides a hierarchical semantic organization of the classes. The idea to use 3D models from the web has an additional advantage: By using 3D models, the problem of coping with a large intraclass variety is inherently addressed, as the number of available models is found to be proportional to the intraclass variety. Scanned objects can also be included into the databases and used with the framework, the choice for using mainly 3D CAD models was due to availability and completeness of the models, as it takes some effort to scan complex objects as chairs for example. 3D CAD models from the web are easily accessible

[1] http://i61p109.ira.uka.de/ObjectModelsWebUI/

[2] http://www.itl.nist.gov/iad/vug/sharp/contest/2010/RangeScans/

[3] 123dapp.com, turbosquid, archive3D.net

but have some drawbacks on their own like non-consistent normal directions, unequal tessellation of triangles and errors in the mesh. Our approach of generating views of the models by rendering them can cope with these errors.

The classes are organized in four increasingly challenging datasets, as more sophisticated descriptors and additional cues are necessary to differentiate between 200 classes in the largest dataset. The according test-databases contain scenes with only a single object per scene. The segmentation is provided as a preprocessing step by the framework.

The four datasets with its properties are introduced in the following sections.

A. *Cat10: Basic Object Classes*

The basic dataset consists of common, simple, geometrically distinguishable but partially similar objects. Object classes were chosen to also be suitable for robotic manipulation. The database consists of 360 3D CAD models in the classes apple, banana, bottle, bowl, car, doughnut, hammer, mug, tetra-pak and toilet-paper. The test-database consists of 1600 scenes of single objects on a flat surface in multiple poses and multiple instances per class. For each scene a color image, a point cloud and a bounding box with the class label is provided. In Figure 3, a representative sample of the Cat10 model and test database is given.

The challenges in these classes are twofold: Firstly the intra-class variance of the classes hammer, mug and bottle, as these three classes are to be found in hundreds of shape variations in the real world. Secondly, the inter-class similarity of the classes (mug,toilet paper),(apple,donut) and (bottle,banana,car) when given only a partial view as depicted in Figure2.

B. *Cat50: Super-Classes*

The Cat50 model database consists of the Cat10 database with forty additional classes. The classes in this database are still distinguishable by shape only, but also include sub-categories (chair, office-chair, armchair and car, convertible, pickup, formula-car). Table I gives an overview of the classes sharing the same hypernym, i.e., belonging to the same superclass.

From the point of view of object classification, organizing objects in a tree has an implicit advantage regarding evaluation: The level of misclassification of an object can be measured as the length between the nodes in the tree. Clearly, misclassification inside a subtree – convertible as car, or airplane as fighter jet – is better than outside a subtree, especially when robustness and user acceptance is of importance as in home robotics.

Fig. 2. Similar partial views of the classes mug vs. toilet paper and and donut vs. apple

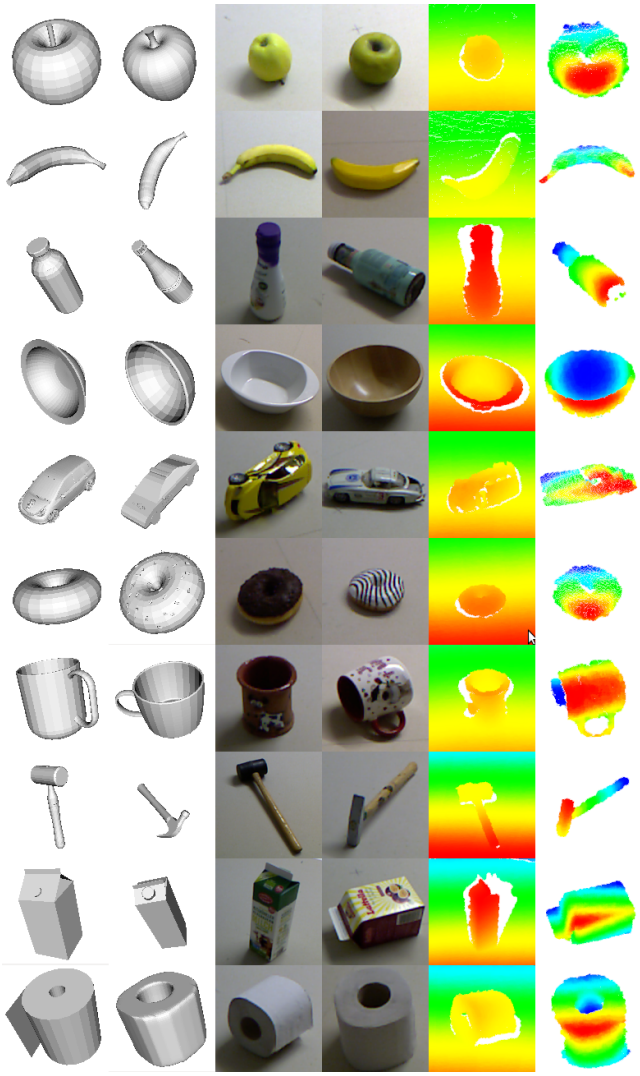


Fig. 3. CAD models of the ten classes with selected test scenes. First two columns present two typical cad-models from the according class followed by two object instances from the testset with the whole scene and segmented scene in point cloud representation.

The according test database for the Cat50MDB adds another 1600 scenes which adds up to 3200 test scenes for the 50 categories.

The challenges in this database include coping with large shape differences although from the same class (paper airplane test objects to CAD model airplanes), similar objects from super-classes and accidentally matching views – as already present in the Cat10 database – as a direct result of scaling the number of CAD models up to 1500. Example views of the challenges are depicted in Figure 4.

C. *Cat100: Color*

This database adds objects which are similar in shape but can be uniquely distinguished when using color as an additional cue. As stated in the work of Lai [12], color together with shape leads to improved recognition of objects and object classes. As depicted in Figure 5, color is not only improving object class recognition, in these one hundred

TABLE I

HIERARCHICAL ORGANIZATION OF THE MODELS IN CAT50.

coarse categories (hypernyms)	shape categories (hyponym)
animal	camel, cow, dinosaur, elephant horse, shark
musical instrument	banjo, guitar
container	bottle, can, mug, tetra pack
edible fruit	apple, banana, lemon, pear starfruit, pineapple, strawberry
motor vehicle	car, convertible, locomotive monster truck, pickup, race car, suv tank, truck
food	donut, pretzel, croissant
aircraft	airplane, biplane, fighter jet, helicopter
seat	armchair, chair, office chair, stool
footwear	boot, sandals, shoe, heels, ski boot,
hand tool	hammer, pliers, screwdriver, wrench

object classes it is crucial to have color as an additional cue to differentiate between the newly added classes. The database now contains many natural objects like fruits and vegetables, which share a common primitive shape such as orange, apple, lemon, lime, watermelon, carrot-radish, etc.

Man made objects are largely excluded from adding to this database, as color can not be assumed fixed, even with common objects such as a tennis ball for example, as it comes in additional colors to the standard yellow.

D. Cat200: Size

One important aspect of objects and object classes was not used and not needed in the previous category databases: size. To successfully distinguish among our 200 categories database, the real world size of the objects becomes important. As classification of objects is subjective – assume a tennis ball with 30 cm diameter, is it still a tennis ball? – we advocate a functional viewpoint on classification: If the object affords the intended function, it is part of the class, otherwise it is a new class, e.g. toy-tennis ball. Following this schematic, a huge part of man-made objects depends on the size cue, e.g., example in Figure 6.

Real world scale information is not yet present in the database, as CAD models do not come with a common real

Fig. 4. Challenges when matching real objects like inflatable and paper airplanes to CAD models of planes which only share overall shape.

Fig. 5. Some classes are almost identical in shape but differ in color. Lemon and lime are two obvious examples, but most roundish shaped fruits and vegetables having color as a distinct cue.

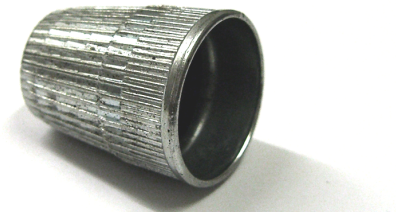


Fig. 6. Size matters: Depending on the real size of the depicted object it can be waste-bin, a mug or a thimble. Shape, color and texture are not sufficient any more to classify this object, which is a thimble.

word size information and therefore it has to be acquired from other resources on the web or learned by the system during successful detections of objects.

E. Community-built Test Database

The test database for these two hundred object classes are open to be extended by the community to provide a large unbiased test database. The test database will be fixed once a minimum of five test objects per category are available for evaluation in the database. We provide tools and web-space for uploading test scenes to 3DNet. A test scene is defined as binary pcd file including X,Y,Z,RGB values captured with a Kinect-like sensor. Capturing can be done using standard PCL tools or our provided ROS-based capturing and annotation tools. To ease segmentation, objects have to be on a flat surface, e.g. on the floor. Annotation is done by 3DNet according to the classes available. For the follow-up database Cat300, classes can be requested by the community via 3DNET.

IV. FRAMEWORK

The proposed open-source PCL-based framework targets real-time classification and object instance 6D0F pose recognition for robotics and provides an easy way of training descriptors, adding new classes or specific objects. Adding new descriptors is supported and encouraged by providing code-templates for an easy transition of C++ code into the framework. Evaluation and benchmarking are also part of the framework, as is 6D0F pose estimation and object recognition.

Usage of the proposed framework for object classification requires the following steps:

1) SVN check-out framework provided on 3DNet
2) Download CAD models from 3DNet
3) Download test database from 3DNet
4) Use present descriptor or implement own one using provided template
5) Run the program to fully automatically train on the CAD models and evaluate on the test sets
6) Plug in a Kinect and classify objects

A. View Generation

The training on CAD models is done by rendering and sampling the z-buffer from views around the model and storing the generated partial views as point clouds. Descriptors are computed on these partial views. The number of

views can be chosen from as few as 12 to several hundreds, depending on the descriptor and application in mind. The standard number of views used for the experiments in this paper is set to 80, as this number provides sufficient views even for complex objects.

Training on 3D models by generating synthetic views has the following advantages:

1) Completeness: Viewpoints are evenly spaces around the object, no missed views
2) Parameterizable: Easily re-trainable with different parameters such as number of views, resolution, noise level or distance to object
3) Sensor independence: Simulation of favorite sensor characteristics possible (field of view, aliasing, noise)
4) Additional information available: Entropy of the views or the connectivity of the views using an aspect graph can be calculated.

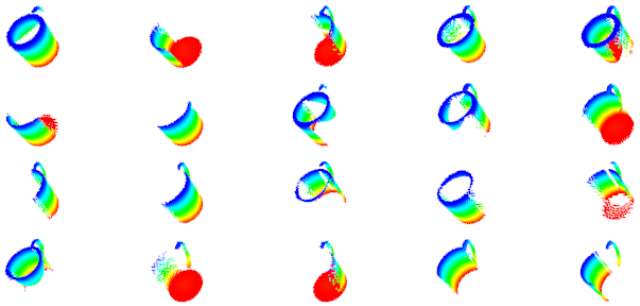


Fig. 7. Partial views of a mug generated by sampling the depth-buffer while rendering views around the object.

B. Entropy

Having synthetic views and the original model at hand enables the calculation of the entropy of each view i.e. the expected value of the information contained in a view. This follows the idea of using the different levels of information in views as shown in [2], where an optimal set of views(images) of a 3D model is found by adaptive clustering.

These entropy values for each view can be used in a post-processing step to filter accidental views: Given a model of a bottle, the view directly from the bottom only represents a small portion of the object and thus has a low entropy value assigned. If this view is matched against something round and curved like an apple or donut, it can be filtered as real world scenes are rarely represent such extremal views of objects.

The entropy is calculated as the ratio of the surface area of the whole model and the visible surface area. Experimental evaluation of view filtering the nearest neighbor list is given in SectionV. Another available post-processing step for filtering is available in the framework using the approach proposed in [24], where the similarity of nearby views is used to filter accidental matches.

C. Pose Estimation

The Camera's Roll Histogram [1] together with any of the descriptors and ICP [3] is used to calculate the pose and the scale of the model and align the 3D model with the scan from the sensor as depicted in Figure 8, which for example, enables robotic manipulation tasks as the full 3D information of the model can be used. The 3D models in the database are not aligned to each other, as robust automatic alignment of 3d models is still an open research field and unsolved given a large intra-class variability. The estimated pose therefore depends on the object's coordinate system.

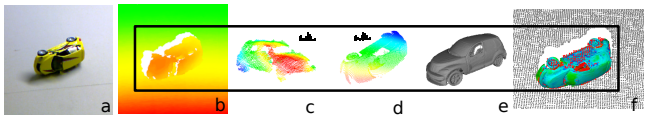


Fig. 8. Pose Estimation: Given a scan from a Kinect (a,b), the segmented point cloud (c) is matched against synthetic views of the model (e). Using the best matching view (d), the model is aligned to the scan(f).

D. Extensibility

Adding a new object class is easily achievable by following the following steps:

1) Download 3D models of new object class from the various sources from the web or scan your objects
2) Convert the 3d models to PLY-format (we suggest meshconv[4])
3) Put the 3d models into a subdirectory of the already existing database and start the framework for view generation
4) Optional: An XML-file in each class directory provides the link to WordNet through the WordNet ID and additional attributes to the class.

E. Descriptors

The proposed framework comes with a set of available descriptors. The choice of descriptors is based on speed, availability and stability. Therefore global 3D descriptor are the first to be entering the framework such as VFH, CVFH, SHOT and shape distributions based descriptors as these provide the needed speed for robotics applications. Reimplementation and adaptation of Spherical Harmonics [10] and Spin Images [9] as global descriptors are the next to be put into action. Local descriptors with Bag-of-Words approaches as used in [5], [13] and [8] require an extra step of learning the visual-words which is not yet available in the framework.

1) VFH: The Viewpoint Feature Histogram (VFH) introduced in [17] is a viewpoint global descriptor based on angular normal distributions extracted from the surface normals and a reference coordinate system obtained by averaging the normals and points on the whole surface. It was designed to robustly describe the geometry of objects seen from a certain viewpoint using the same depth sensor for training and detection. The average time for calculation

[4]http://www.cs.princeton.edu/ min/meshconv/

and matching is approximately 70 ms where the normal calculation is accountable for most of the calculation time.

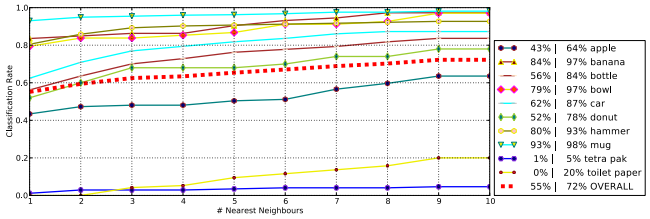


Fig. 9. VFH rank plot on the 10 classes test database against 10 Classes. VFH produces good results but fails on two classes.

2) CVFH: The Clustered Viewpoint Feature Histogram [1] is a semi-global view based descriptor based on VFH. Because of its semi-global nature, only certain parts of the objects are used to build the reference systems on which the computation is based but uses the whole available view information to build the angular normal distribution histograms. Because of its multivariate representation of a partial view, it can deal with partial occlusions and cope with different data characteristics between training and detection. The parts of the object used to build the coordinate systems are obtained by a smooth region growing stage aiming to detect stable regions which are robustly estimated by the depth sensor. The descriptor computation time depends strongly on the region growing step, both in the number of points and the number of stable regions found. The average time for computation and search is approx. 208 ms, ranging from 50 ms and 300 ms.

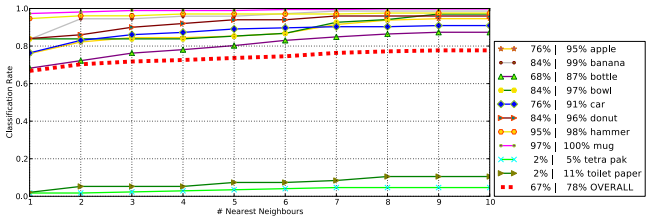


Fig. 10. CVFH rank plot shows improvement over VFH on 10 classes, but also has problems with two classes.

3) SDVS: The Shape Distribution on Voxel Surfaces descriptor is a descriptor based on histograms of point-to-point distances and was introduced in [22]. The point distances are classified to be either on the surface of the partial view, off or mixed. This descriptor is calculated directly on the point cloud and does not need any normals to be computed and takes an average of 25 ms for calculation and matching.

4) ESF: The Ensemble of Shape Functions descriptor introduced in [23] is based on the SDVS descriptor and includes multiple shape function as described in Osada [14], such as A3(angles), D2(lengths) and D3(areas) which increases the classification performance. The descriptor requires 45 ms for calculation and matching. A supplemental video representing 3DNET of classifying object with this descriptor is available on 3DNet (3d-net.org/video).

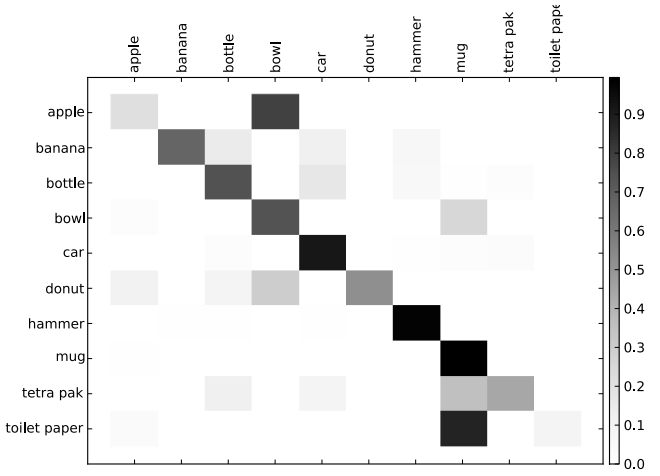


Fig. 11. SDVS rank plot on the 10 classes test database against 10 Classes. Most confusion is between the classes mug and toilet paper and between bowl and apple as partial views of these classes resemble parts of the other class.

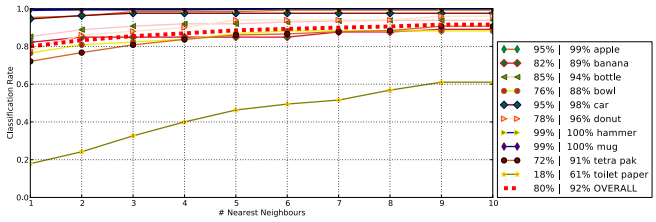


Fig. 12. ESF rank plot on the 10 classes test database against 10 Classes. The tetra pak class is working for this descriptor, but it still has problems with the similarity of mug and toilet paper classes.

5) SHOT: The SHOT descriptor introduced in [20] is aimed at surface matching with local descriptors, but is used here as a global descriptor for the whole object. The descriptor showcases a high classification rate, but compared to the other approaches the calculation time is up to a magnitude larger, so the feature calculation and matching takes from 130 ms to 400 ms on our test database.

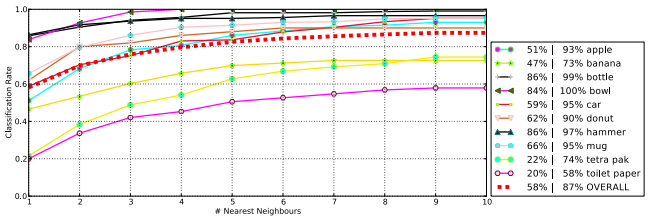
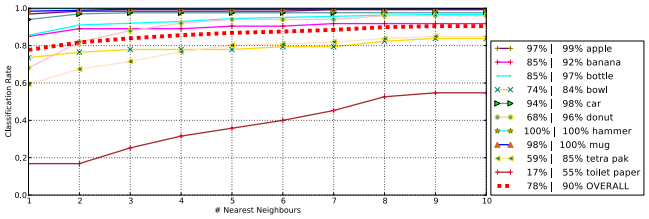


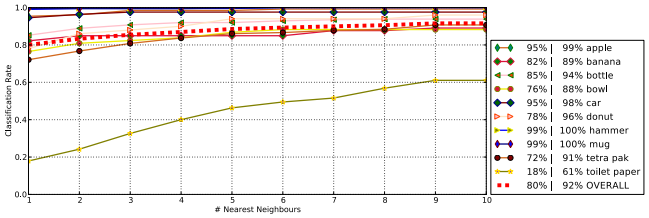
Fig. 13. SHOT rank plot on 10 classes provides good results with no class less then 20%, but is also the slowest descriptor in this benchmark.

F. Weight Learning on Synthetic Views

Parameters and descriptor weights can be learned on the synthetic views without having to see a single real scene. The improvement of the descriptor performance is showcased in Figure 14 where a 2 % improvement was achieved by learning the descriptor sub-histogram weights on a sample



(a) ESF descriptor with equal weights, no training.



(b) ESF descriptor with learned weights from synthetic views shows 2 % performance improvement.

Fig. 14. Weights learned on synthetic views for increased classification rate.

of the synthetic views. This method for tuning descriptors can be accomplished with any descriptor having sub-parts in its histogram and therefore weights can be learned. The big advantage here is that this can be done offline, without having a test database to split in training and evaluation parts.

V. BENCHMARK & EVALUATION

3DNET's intention is to provide benchmarks for 3D shape descriptors on the test databases in a similar way the Middlebury Stereo Benchmark [19] is for dense stereo.

For every descriptor rank-plots, confusion matrices and overview statistics are generated for the test sets against the model databases, e.g. 10-10, 10-50, 10-200, to provide insight and conclusions on descriptor performances. A sample benchmark is given for the ESF descriptor for 200 classes in Figure 15 and Table II.

As speed is a key issue in addition do classification performance for robotics, we do not follow the approach of the Middlebury Benchmark providing user to submit benchmarks. To foster sharing open-source code and enabling comparable performance measures, users are invited to include their descriptor in the framework, add test scenes to the test databases and add new categories, but benchmarking and providing the results on 3d-net.org is done by 3DNet itself.

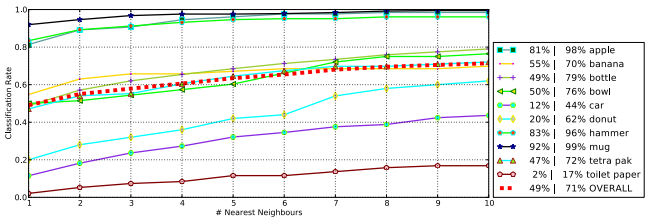


Fig. 15. ESF rank plot on Cat10 test database against 200 Classes.

VI. CONCLUSIONS

A novel methodology is presented for rapid and scalable training of 3D shape descriptors using CAD models. To accomplish objective comparison of shape descriptors, 3DNet (3d-net.org) is presented as a free resource providing an open-source framework and test databases for benchmarking. Model databases with CAD models in 10,50,100 and 200 categories are presented as a common training resource. 3DNet offers to be extended by the community by adding new categories, creating a common test database and sharing new shape descriptors. 3DNet provides all necessary resources to process scenes captured with a Kinect style camera as depicted in Figure 16. At the current state, segmentation is the main performance bottleneck, detaining us from having higher frame-rate classification. This leaves a lot of scope for future improvements in the challenging areas of handling touching objects, occlusions and sensing inadequacies and speed. We hope that with 3DNET, progress in these specific areas and in 3D object and object class recognition is accelerated.

Fig. 16. Classification using ESF with nearest neighbor on a scene with multiple objects. Challenges are wrong and missing segmentations, sensor noise and missing data on shiny and transparent objects and parts and descriptor flaws, which cause mis-classification. As this classification is done per frame, using multiple, successive frames and/or alignment of 3D models to sensor data, wrong classifications can be filtered as a post-processing step.

TABLE II
NEAREST NEIGHBOR CLASSIFICATION AND MOST CONFUSING CLASS

class name	1-NN	10-NN	confusing class
per scenes OVERALL	58.22 %	78.23 %	
per class OVERALL	49.10 %	71.39 %	
apple	81.40 %	98.45 %	pumpkin
banana	54.79 %	69.86 %	pistol
bottle	48.77 %	79.01 %	suv
bowl	50.00 %	76.47 %	hat
car	11.52 %	43.64 %	suv
donut	20.00 %	62.00 %	cap
hammer	83.41 %	96.10 %	axe
mug	91.96 %	99.46 %	watch
tetra pak	47.09 %	72.09 %	mug
toilet paper	2.11 %	16.84 %	armchair

REFERENCES

[1] A Aldoma, N Blodow, D Gossow, S Gedikli, R B Rusu, M Vincze, and G Bradski. CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues. *3rd IEEE Workshop on 3D Representation and Recognition*, 2011.

[2] Tarik Filali Ansary, Mohamed Daoudi, and Jean-Phillipe Vandeborre. 3D Model Retrieval based on Adaptive Views Clustering. In *International Conference on Advances in Pattern Recognition (ICAPR)*, 2005.

[3] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.

[4] J Deng, W Dong, R Socher, L.-J. Li, K Li, and L Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition*, 2009.

[5] H Dutagaci, A Godil, C P Cheung, T Furuya, U Hillenbrand, and R Ohbuchi. SHREC 2010 - Shape Retrieval Contest of Range Scans. In *Eurographics Workshop on 3D Object Retrieval*, 2010.

[6] Christiane Fellbaum. WordNet: An Electronic Lexical Database. *Cambridge, MA: MIT Press*, 1998.

[7] Takahiko Furuya and Ryutarou Obuchi. Dense Sampling and Fast Encoding for 3D Model Retrieval Using Bag-of-Visual Features. In *ACM International Conference on Image and Video Retrieval*, pages 0–7, 2009.

[8] C Goldfeder, M Ciocarlie, J Peretzman, Hao Dang, and P K Allen. Data-driven grasping with partial sensor data. In *International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[9] Andrew E Johnson and Martial Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligences*, 21(5):433–449, 1999.

[10] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. *SGP*, pages 156–164, 2003.

[11] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A Large-Scale Hierarchical RGB-D Object Dataset. *International Conference on Robotics and Automation (ICRA)*, 2011.

[12] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Sparse Distance Learning for Object Recognition Combining RGB and Depth Information. *International Conference on Robotics and Automation (ICRA)*, 2011.

[13] R Ohbuchi and T Furuya. Scale-weighted dense bag of visual features for 3D model retrieval from a partial view 3D model. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 63–70, 2009.

[14] R Osada, T Funkhouser, B Chazelle, and D Dobkin. Matching 3D models with shape distributions. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 154–166, May 2001.

[15] Morgan Quigley, Ken Conley, Brian P Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[16] B.C. Russell, Antonio Torralba, K.P. Murphy, and W.T. Freeman. LabelMe: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[17] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3D recognition and pose using the Viewpoint Feature Histogram. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2155–2162, 2010.

[18] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *International Conference on Robotics and Automation*, Shanghai, China, 2011.

[19] D Scharstein and R Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002.

[20] Frederico Tombari, Samuele Salti, and Luigi Di Stefano. Unique Signatures of Histograms for Local Surface Description. *11th European Conference on Computer Vision*, pages 347–360, 2010.

[21] Antonio Torralba and Alexei A Efros. Unbiased Look at Dataset Bias. *IEEE Computer Vision and Pattern Recognition*, 2011.

[22] W Wohlkinger and M Vincze. Shape Distributions on Voxel Surfaces for 3D Object Classification From Depth Images. *IEEE International Conference on Signal and Image Processing Applications*, 2011.

[23] Walter Wohlkinger and Markus Vincze. Ensemble of Shape Functions for 3D Object Classification. In *International Conference on Robotics and Biomimentics*, 2011.

[24] Walter Wohlkinger and Markus Vincze. Shape-Based Depth Image to 3D Model Matching and Classification with Inter-View Similarity. *International Conference on Intelligent Robots and Systems*, 2011.