Project Acronym:	GRASP
Project Type:	IP
Project Title:	Emergence of Cognitive Grasping through Introspection, Emulation and Surprise
Contract Number:	215821
Starting Date:	01-03-2008
Ending Date:	28-02-2012

Deliverable Number:	D21
Deliverable Title :	Integrated perception/context model extended to a grasping ontology to contain cues related to affordances
Type (Internal, Restricted, Public):	PU
Authors	M. Vincze, W. Wohlkinger, A. Aldoma, K. Varadarajan, E. Potapova, D. Fischinger, S. Olufs, M. Zillich; D. Gonzalez-Aguirre, J. Hoch, S. Röhl, T. Asfour; M. Pronobis, D. Kragic;
Contributing Partners	TUW, KIT, KTH

Contractual Date of Delivery to the EC:	28-02-2011
Actual Date of Delivery to the EC:	28-02-2011

Contents

Chapter 1

Executive Summary

This report presents the work of year three in WP4. WP4 is concerned with perceiving the object and hand involved in the grasp and all contextual information relevant. With grasp context we refer to the information relevant to the grasp, which at its core includes the grasp points on the objects but also the relationship to the complete object, the hand, the task, and the attention on the target object. The overall objective is to perceive grasping points on unknown objects by the end of the project.

Work in year three concerned

- **[Task 4.2] - Perceiving task relations and affordances** The objective is to exploit the set of features extracted in Task 4.1 to obtain a vocabulary of features relevant to the grasping of objects and to learn the feature relations to the potential grasping behaviours and types.
- **[Task 4.3] - Linking structure, affordance, action and task** The objective is to provide the necessary input to the grasping ontology developed in WP2, which represents knowledge about the grasping experiences learned. It contains relations and constraints to (1) the object and its properties such as size, shape and weight, to (2) perceived affordances (potentialities for actions) and grasping points, to (3) the task that is executed, e.g., grasping for pick up or to move as cup, and to (4) the context or surrounding of relevance. It is investigated how such a link can be efficiently established and used to obtain task-based grasping of object categories and to achieve extendibility for grasping new objects.

The work in this deliverable relates to the following second year Milestones:

- **[Milestone 10]** Linking structure, affordances, actions and tasks and a first evaluation of representations defined by the ontology.

The advance in year three focused on learning object categories to generalise grasps to new objects in relation to known object classes.

- **Evaluating features for 3D object class detection:** This work uses previous results from WP5 to merge stereo views and obtain a more complete 3D image of objects on a table for classification. A shape model based approach and machine learning are used for object categorization has been implemented and tested on ARMAR-IIIa. Visual sensing from different view points allows the reconstruction of 3D mesh-models of the objects found in the scene by exploiting knowledge about the environment for model-based segmentation and registration. These reconstructed 3D mesh-models were used for shape feature extraction for categorization and provide sufficient information for grasping and manipulation. Finally, visual categorization was performed with a variety of features and classifiers allowing properly categorisation of unknown objects even when object appearance and shape differ from the training set. The approach is evaluated with ARMAR moving around a table with a single object to merge views obtaining a nearly complete 3D model and then categorise the model. 35 objects belonging to 8 categories have been tested. Details are presented in Appendix [A].

- **Learning object classes from the web for single view detection:** The goal is to have a robot classify never before seen objects within a single view in a fast and robust manner. Instead of taking many images in a large database, we exploit Web resources such as 3D-Warehouse to obtain a model of object classes from the models of many individuals of a class. The work shows how view-based data from the 3D Web models is used to efficiently train for such a single view detection. The classification task itself is seen as a matching problem, finding the most appropriate 3D model and view to a depth image. We show that a single view using an RGB-D sensor is sufficient to classify a novel object. To achieve robust yet fast classification, we use an ensemble of state-of-the-art classifiers that directly operates on the 3D points of the sensor without any calculation of normals or generating a mesh from it. The approach requires a first segmentation that is achieved with methods from year two. To move towards grasping objects out of a box we work on improving these methods, since at present segmentations either over or under segment, which is not sufficient to obtain reliable results for grasp point detection. The result of the classification is a labelling of the image region of interest. However, the classification approach itself does not deliver accurate pose information of the object. Hence, in a further step of the object classification process, object pose needs to be determined. Appendix [B] presents details of this work.

- **Aligning class models to obtain object pose for grasp point determination**: Here we introduce a novel method for the pose alignment of geometrically similar 3D models. Similarity has been already achieved in the previous step through object classification. Pose alignment is based on the prior that both models have at least one common tangent plane on which both can stand stably and when standing on it the models are partially aligned. The use of such a "common sense" rule greatly simplifies and hence robustifies the remainder of the procedure. Furthermore, object pose is linked to the typical object affordances, hence this enables to link affordances with pose and object shape in a formal way (see WP2). It is only necessary to determine the final rotation around the normal of this stable plane. For this we use an image alignment technique based on the log-polar transformation. Of particular interest to the approach of using one view only is that the method does not rely on any kind of global symmetry features. Hence we show it can be used to register incomplete stereo point clouds of objects located on a stable plane (table, ground, etc.) with the corresponding similar 3D models. We evaluated the method by aligning 120 models belonging to 12 different classes and a comparison to state-of-the-art methods. Appendix [C] presents details in form of a recently accepted paper. This approach has also been used to enable pose related task learning in WP2.

- **Part-based grasp points:** Results in the first two years [Task 4.1] showed that local image information can be very well used to obtain shape information about objects. Based on this, a new method for learning grasp points in relation to object parts is investigated. The idea is to link object parts with the affordances and tasks formulated in WP2. This enables to break down the detection of new object to object parts, which in themselves typically indicate where to grasp an object. We attempt to extend the scope of affordance features to define Conceptual Equivalence Classes and to recognize these classes leading to scalable unit (part/ part assembly/ object) recognition system. The advantage is that grasp points from related object classes can then be used for grasping of new objects. First work towards this goal is presented in Appendix [D].

- **Object classification from 2D and 3D data:** Grasping knowledge can be transferred between objects that belong to the same object category due to their similar geometric properties and functionality. Moreover, given a specific task we can generate a set of the most suitable grasps for each object category. For example, when pouring from a cup it should be grasped by its handle not from the top. We developed an object category recognition system that employs 2D (RGB image) and 3D (point cloud representing a partial view of an object) information about an object and several strategies for integrating 2D and 3D information. The system was evaluated on real data collected using active stereo cameras and for a number of household object categories obtaining a high recognition rate. The system was integrated with the active segmentation module already used for the year two demonstration and the task-constrained grasp planning system (WP2) constituting a comprehensive system that generates a set of suitable grasp for objects in a natural scene specific to their categories and constrained by the performed task. Details are given in Appendix [E].

Appendix A

Appendix A: Attached Papers

A D. Gonzalez-Aguirre, J. Hoch, S. Röhl, T. Asfour, E. Bayro-Corrochano, R. Dillmann: Towards Shape-Based Visual Object Categorization for Humanoid Robots; IEEE ICRA 2011, accepted.

B Walter Wohlkinger, Markus Vincze: Towards Robust Shape-Based Depth Image to 3D Model Matching using Inter-View Similarities; prepared to be submitted to IEEE IROS 2011.

C Aitor Aldoma, Markus Vincze: Pose Alignment for 3D Models and Single View Stereo Point Clouds Based on Stable Planes; 3DIMPVT - The First Joint 3DIM/3DPVT Conference 3D Imaging, Modeling, Processing, Visualization, Transmission; 2011, accepted.

D Karthik Varadarajan, Markus Vincze: Affordance based Part Recognition enabled Visual Cognitive Engine for Grasping and Manipulation; prepared to be submitted to ICAR 2011.

E Marianna Pronobis, Danica Kragic: Combining 2D and 3D object categorization for task constrained grasping; prepared to be submitted to IEEE IROS 2011.

Towards Shape-Based Visual Object Categorization for Humanoid Robots

D. Gonzalez-Aguirre, J. Hoch, S. Röhl, T. Asfour, E. Bayro-Corrochano* and R. Dillmann
Karlsruhe Institute of Technology, Adenauerring 2, Karlsruhe-Germany.
{gonzalez, julian.hoch, roehl, asfour, dillmann}@ira.uka.de
**CINVESTAV-Unidad Guadalajara, Av. Científica 1145, Mexico*
edb@gdl.cinvestav.mx

***Abstract*— Humanoid robots should be able to grasp and handle objects in their environment, even if the objects are seen for the first time. A plausible solution to this problem is to categorize these objects into existing categories with associated actions and functional knowledge. So far, efforts on visual object categorization using humanoid robots have either been focused on appearance-based methods or were restricted to object recognition without generalization capabilities.**

In this work, a shape model based approach using stereo vision and machine learning for object categorization is introduced. The state-of-the-art noise-tolerant shape-matching and shape-retrieval features were evaluated and selectively transfered into the visual categorization task. Visual sensing from different vantage points allows the reconstruction of 3D mesh-models of the objects found in the scene by exploiting knowledge about the environment for model-based segmentation and registration. These reconstructed 3D mesh-models were used for shape feature extraction for categorization and provide sufficient information for grasping and manipulation. Finally, the visual categorization was successfully performed with a variety of features and classifiers allowing properly categorization of unknown objects even when object appearance and shape substantially differ from the training set. Experimental evaluation with the humanoid robot ARMAR-IIIa is presented.

I. Introduction

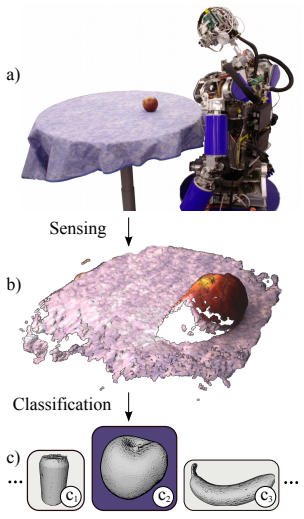


Fig. 1: Shape-based visual object categorization. a) The humanoid robot ARMAR-IIIa is exposed to an unknown object. b) Visually reconstructed scene. c) Object categories represented by shape-models from the training set.

In everyday scenarios, humanoid robots need to interact with a wide variety of different objects in the environments in order to perform complex tasks. Reliably recognizing these objects is a problem which has been actively researched. As a consequence there are now well-established appearance-based methods for the task of object recognition [1][2] where only previously learned objects have to be recognized.

However, an autonomous humanoid robot also needs to deal with objects it has never encountered before. This generalization skill is the purpose of object categorization, where the encountered objects are assigned to previously defined categories. These categories are *basic level classes* which can be mostly characterized by their shape, see [3]. For example, object instances of the category *bottles* have different shape, size, texture and color, still the robot should be able to correctly categorize a bottle even if it has not seen this particular exemplar before.

Furthermore, categories like *fruits* or *vegetables* change their appearance (color and texture) with progressing ripeness but maintain their overall shape. By using shape-based representations it is possible to simultaneously deal with these circumstances and supply the essential grasping and manipulation information.

In this approach, a set of predefined object categories with several training samples was created. The training samples consist of labeled visually reconstructed 3D mesh-models for each category. Afterwards, a variety of classifiers was trained using shape features extracted from these 3D mesh-models. Subsequently, in the on-line phase, the acquired 3D meshes with the stereo camera of humanoid robot were used to extract shape features. Finally, the trained classifiers were applied to categorize unknown objects, see Fig.1.

II. Related work

The focus of this work is to properly categorize small, rigid and graspable objects typically found in a human household environment while coping with the challenges of the limited visual sensing capabilities such as sensor's dynamic range, resolution, noise and self-occlusion.

Classic approaches for object recognition in robot vision solely use the object appearance [4] and [5]. Besides, in order to be able to manipulate an object, the 6D pose has to be determined. While it is possible to get the object pose with appearance based methods when using the depth information from stereo cameras [6], a more common approach is to match stored 3D models to the scene [7].

Appearance-based categorization approaches include the *Bag-of-Features* [8] methods, which determine the distribution of local features in the feature space and the part-based approaches [9], which model objects as a collection of image parts or features.

When objects of different categories only differ in shape and not in texture (for example a wooden saltshaker and a wooden trivet), appearance-based methods quickly reach their limits. With a model-based approach, the object's 3D shape can be incorporated into the categorization process.

In [10] and [11], point clouds were obtained from the objects using a structured light projection and stereo camera on a mobile robot. The Fast Point Feature Histograms

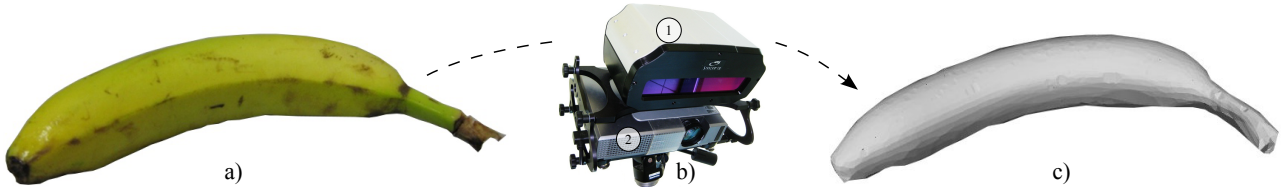


Fig. 2: The off line 3D mesh-models acquisition for training. a) Training sample object(s). b) The system used to digitize the training objects: [1]-Stereo camera, [2]-Projector. c) The digitized 3D polygon representation.

were calculated from surface points and the object were classified using Support Vector Machines and Conditional Random Fields. They achieved accurate results (96.69% category identification, although they used the same objects for training and testing), but use only four categories with little inter-class variance. However, since most humanoid robots do not have light projectors, this sensing approach is not a viable option for on-line categorization systems using humanoid robots.

Another common approach is to use *spin image* features, which describe the global shape of the object from the perspective of local points on the object's surface. Spin images are shape descriptors which have been applied to surface matching [12], object recognition [13] and [14], 3D registration [15] and 3D object retrieval [16]. In [17], objects were modeled as consisting of three parts which were categorized by spin images using the recognized part classes. The input data consists only of noise-less simulated laser scanner point clouds, achieving good results categorizing cars into eight categories. In [14], spin images are used in a 3D object detection system with the humanoid robot HRP-2 [18]. The scene was captured with stereo cameras and converted into a point cloud for the 3D mesh construction. Random scene points were selected and the corresponding spin images were calculated. These points were matched to previously calculated spin images of the model to be localized. This approach was designed to find a known object in the scene and does not deal with generalization such as the categorization of unknown objects.

Among the many features that have been used for 3D model-based object recognition are *tensors* [19], *spherical harmonic representations* [20], *shape distributions* [21], *coarse filters* [22] and *conformal factors* [23]. Only three of those features degradate gracefully when dealing with occlusion and sensor noise expected in real applications. Because of their promising properties and superior noise degradation, i) spin images, ii) shape distributions and iii) coarse filters were selected to visually categorize objects by the humanoid robot through stereo vision.

In addition, among the different ways to represent objects with 3D models, like point clouds, voxel representations, octrees or collections of primitives like boxes [24], the 3D polyhedron models were selected due to the efficient construction and calculation of a wide variety of features. There are different algorithms to reconstruct polyhedral from point clouds, the most prominent being the *power crust* [25] and the *tight cocone* [26].

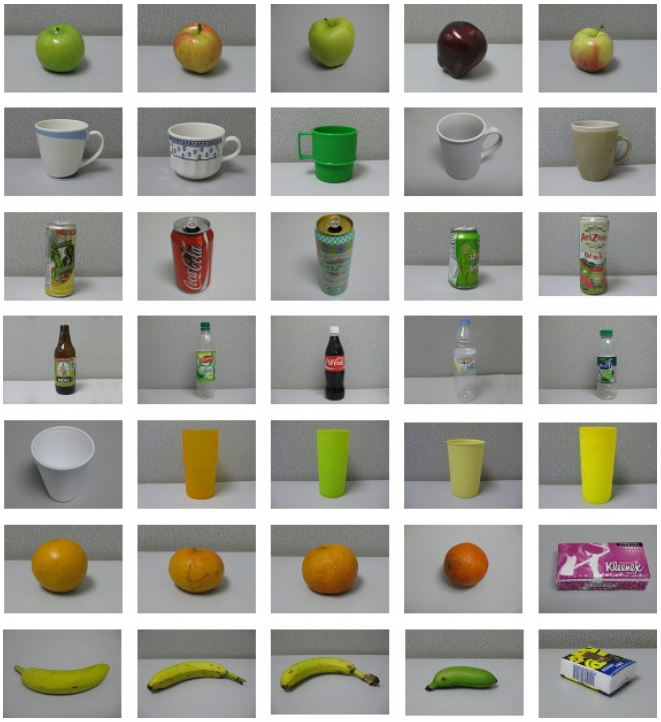


Fig. 3: The training data set. Objects for each category with different shapes and sizes were selected to capture the natural intra-class variance.

III. Methodology

The categorization is based on supervised learning to infer from known training samples to unknown observed objects. Classifiers are trained on a set of labeled training samples and applied to the objects that the robot encounters in its environment. The approach consists of the following phases:

- *Training*: Create training data and train categories.
- *Acquisition*: The robot gathers, segments and registers images of an unknown object to be categorized.
- *Reconstruction*: Obtain 3D meshes from the images.
- *Categorization*: Manages the shape feature extraction and the evaluation of the trained classifiers.

A. Training

The training of reliable classifiers requires sufficiently large database of labeled training objects. Although there are public databases available with labeled object models, like the Princeton Shape Benchmark [27] or the KIT ObjectModels Web Database [28], these were not suitable.

Often, like in the Princeton Shape Benchmark, the models represent artificial objects and are simplified representations of real objects. Therefore, they do not contain the real object dimensions, which can carry important information for the categorization process.

Other databases contain high quality and dimension-retaining representations of real objects, but unfortunately they lack the necessary variety of shapes needed for classifier training. The KIT object models Web Database is more geared towards appearance-based approaches and disposes of a large variety of box-shaped and cylinder-shaped models, but for other shapes there are only single models available.

Due to these limitations, a new training database was created, see Fig.3. This model database comprises 35 objects belonging to 8 categories with sufficiently different shapes: *apples*, *mugs*, *beverage cans*, *oranges*, *bottles*, *bananas*, *beakers* and *tissue packages*. Notice that some categories with similar shapes were chosen (like *apples* and *oranges*), to determine if the small differences between the categories are discriminative enough, especially in presence of larger intra-class variance.

The 35 selected training objects were scanned using a StarCam™3D camera system. It projects structured light on the object and captures the resulting patterns with a stereo camera, see Fig.2. It densely samples the surface of an object from different angles to create a 3D reconstruction.

For each object, a 3D point cloud was obtained (with approximately 5000 points) and a watertight surface representation created using the power crust algorithm, see Fig.2-c. The resulting meshes consist of approximately 10,000 to 15,000 convex polygons, which is small enough for fast feature extraction in less than one second. This database of real world objects was used to extract discriminant features in order to train different classifiers, see Sec.III-D.

B. Visual Acquisition

In the on-line evaluation, the humanoid robot ARMAR-IIIa (see Fig.1) attains the 3D object reconstruction. Since from one point of view only a part of the object is visible, the object is circumnavigated by the robot and several stereo views of the object were captured, see Fig.4. These views were used to create a 3D surface model of the object.

Because the stereo reconstruction is sensitive to lighting effects such as over-exposure, under-exposure and gloss, additional image preprocessing is performed prior to the actual reconstruction step. For each view, several images with different shutter speeds are captured and combined to create a tone mapped HDR image [29], which improves the image quality and preserves local contrast, see Fig.5.

Finally, the object to recognize is segmented in the registered input images using a model-based environmental segmentation algorithm, which exploits the CAD-model of the table and the relative pose of the humanoid robot during the acquisition, see author's previous work in [29].

C. Reconstruction

For each set of images captured from one specific position, a point cloud of the scene is calculated using stereo reconstruction. For correspondence analysis, an extension of the Hybrid Recursive Matching method is used [30].

In two-stage process, the block recursion and the pixel recursion step calculate a new disparity value for the current

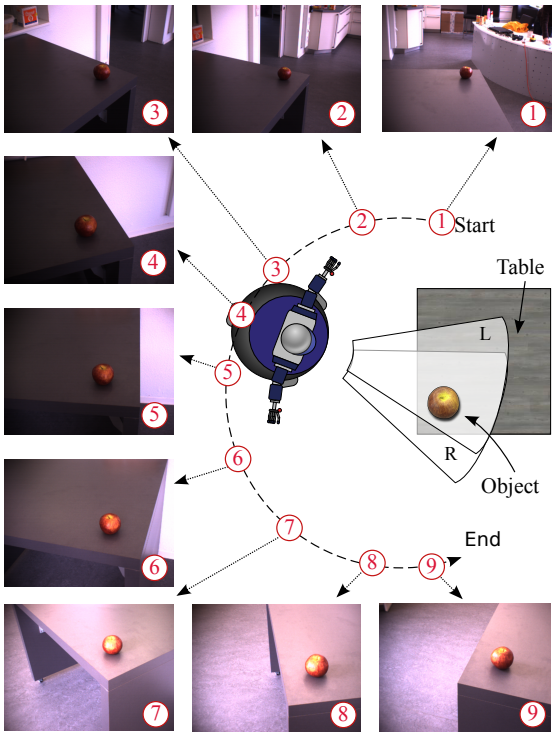


Fig. 4: The acquisition path around the test objects was defined by specifying a start and end point and several way points in between. The humanoid robot ARMAR-IIIa calculated an interpolated path between the given positions and traversed it, making several stops to capture stereo images from the object from different perspectives.

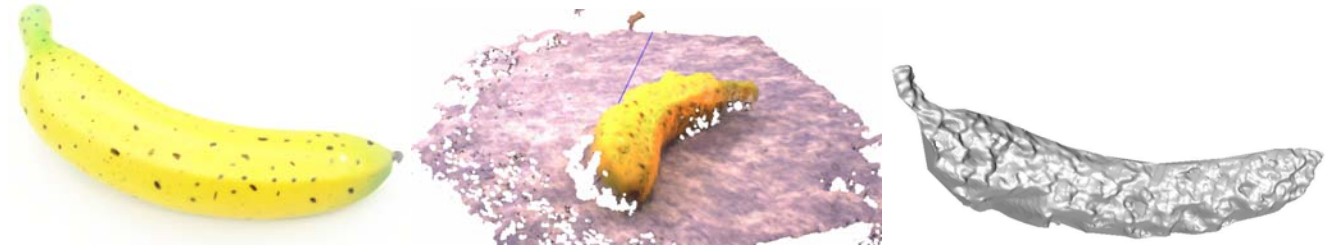
pixel by choosing between four different candidates in the other image. For each candidate, the similarity is calculated using block matching. The with the candidate most similar neighborhood is set as the correspondence. While the block recursion step ensures a smooth disparity distribution (especially in low textured regions), the pixel recursion step introduces new values in regions of discontinuity.

The outliers detection is done via a consistency check where disparities of the left and right disparity image are compared and rejected if their difference exceeds a threshold. Finally, the 3D coordinates of the image points are calculated by using the detected correspondences.

The resulting point clouds of the individual views were then fused using environmental visual cues (known table edges and region growing segmentation), and the final point cloud was preprocessed to remove outliers by calculating the object's centroid and the mean distance $\bar{d}$ between points belonging to the object and the centroid. Points for which the calculated distance d was much greater than the mean distance ($d > \theta \cdot \bar{d}$ with θ set to 2.5) were removed from the point cloud. The resulting point cloud was converted into a surface mesh using the power crust algorithm, see Fig.6. The tight cocone algorithm [26] was also evaluated, however, power crust provided models with less noise artifacts.

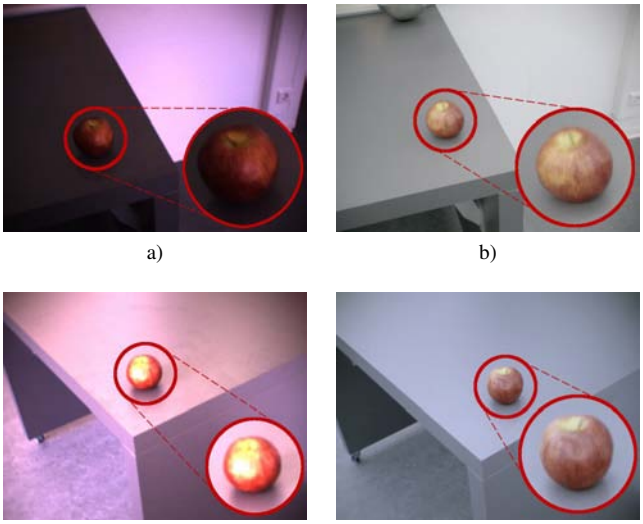
D. Categorization

Finding a good set of features is crucial for the object categorization task. Adequate features should have high discriminative power and should be robust to noise and other sources of variation. They also need to be efficient, pose invariant and capable of partial matching [31].



a) b) c)

Fig. 6: 3D Reconstruction of a banana by the humanoid robot stereo vision. a) Original object. b) Calculated point cloud. c) Reconstructed surface mesh.



a) b) c) d)

Fig. 5: HDR image acquisition. Captured images exhibit areas of a) under- and c) over-exposure. Tone-mapped HDR images also show details in b) very dark or d) very bright areas.

Due to these criteria, spin images, shape distributions and coarse filters were selected to perform shape categorization.

1) Spin Images: The performance of spin images is influenced by three generation parameters:

- *Image size*: s determines the size and resolution of the spin image and the number of bins s^2.
- *Bin size*: b sets the distance between the different bins and determines the support distance $d = s \cdot b$. Increasing d results in a more global behavior of the spin images.
- *Support angle*: determines if the object's rear side is also considered for the calculation. Increasing the support angle leads to a more global behavior.

Spin images were used as features for object recognition by training standard classifiers like support vector machines and artificial neural networks. In the categorization phase, a number of scene spin images is classified and the object that receives the most votes is selected. Notice that dimensionality reduction using PCA of the spin images was performed prior to training.

2) Shape Distributions: Shape distributions [21] are histograms of a *shape function* that cover geometric properties of an object. Possible shape functions are for example the distance between two random points on the object surface or the angle between three random surface points, see Fig.7. The shape function is evaluated for many random samples

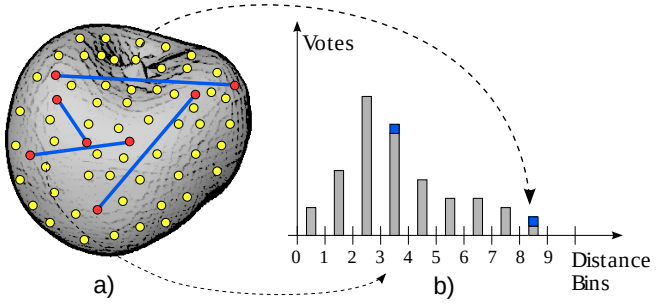


Fig. 7: Fundamental concept of distance histogram calculation [21]. a) Object surface mesh with sampled surface points (yellow). Distance calculation for randomly selected point pairs (red). b) Distance distribution calculation.

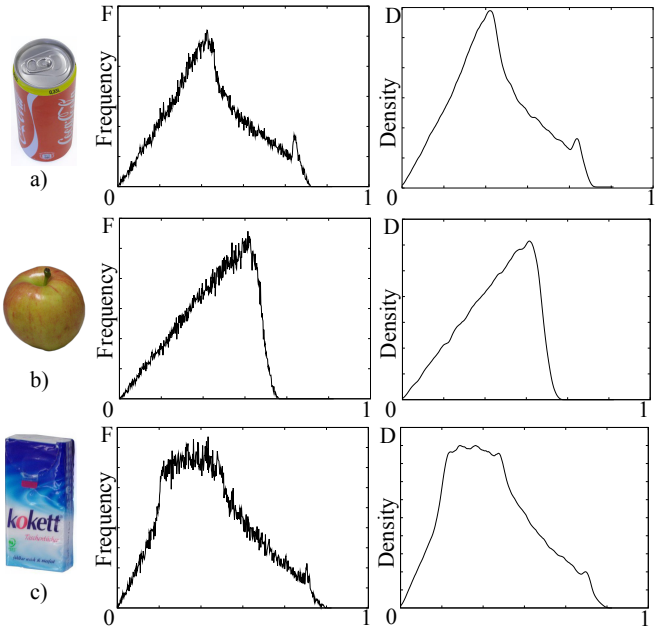


Fig. 8: The distance histogram and the estimated pdf using kernel density estimation. a) Cylindrical object. b) Spherical object. c) Box object.

and the resulting histogram can be used for matching using dissimilarity metrics or standard classifiers.

In the presented work, the D2 measure was implemented, the distance between two random surface points (also called distance distribution), which yields the best categorization results according to [21]. In order to obtain a smooth and stable histogram, kernel density estimation was applied with an Epanechnikov kernel because of its performance and theoretical properties, see Fig.8.

3) Coarse Filters: Coarse filters are features that are created by calculating geometric properties of a 3D model. In

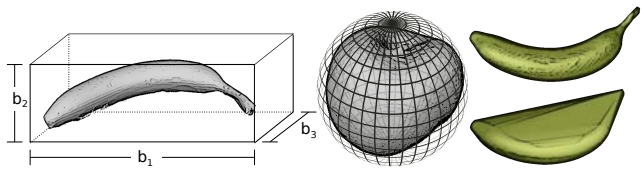


Fig. 9: Concepts used for coarse filter calculation. a) Bounding Box. b)Bounding Sphere. c) Convex Hull.

Volume and Surface Area	Volume V, Area A
Bounding Box	Sides $b_1, b_2, b_3 (b_1 \geq b_2 \geq b_3)$ Ratios $b_1/b_3, b_2/b_3$
Cuboid Ratio	$V/(b_1 \cdot b_2 \cdot b_3)$
Bounding Sphere	Radius r_{bs}, Volume V_{bs}
Sphere Ratio	V/V_{bs}
Convex Hull	Volume V_{ch}, Area A_{ch}
Convexity	A/A_{ch}
Compactness	V/V_{ch}
Hull Packing	$1 - V/V_{ch}$
Hull Compactness	A_{ch}^3/V_{ch}^2

TABLE I: Features in the coarse filters feature vector, see Fig.9.

[22] this approach is used for a shape matching engine with measures like volume, surface area, volume-to-surface area ratio, bounding-box aspect ratio (longest to shortest edge) and some derived values including the surface and volume of the object's convex hull.

In this work, 17 features were tailor into a feature vector. They included the dimensions of the bounding box, the bounding sphere, object area and volume, area and volume of the convex hull, as well as several deduced features like convexity and compactness, see Tab.I.

Taking the calculated feature vectors spin images, coarse filters or shape distributions directly as input for a classifier often leads to inferior results. If some dimensions in the feature vector contain very large values and dominate the others, it is necessary to rescale or normalize the feature vector. The selected classifiers are:

- Soft margin support vector machines with linear kernels and RBF kernels.
- Multilayer perceptrons with one hidden layer.
- K-nearest neighbor classifiers with different values for k and different distance metrics.

In order to estimate the optimal parameters for the different classifiers, a grid search [32] was implemented by a cross-validation performance evaluation on the training database.

IV. Experimental Evaluation

A. Artificial Data

First, the system was evaluated using only the noise-free objects digitized with the 3D scanner in order to evaluate the pure discriminative power of the classifiers.

First, the classifiers were applied them to the training set. Each of the experiments was performed by applying a modified cross-validation approach, namely, randomly separating the database into a subset of training and a subset of test samples, with the training sample ratio being 3:1. Later on, a classifier is trained on the training samples and evaluated

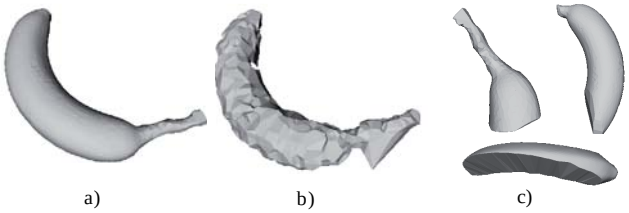


Fig. 10: Artificial model deterioration. a) Original model. b) Noisy model. c) Partially occluded object.

on the test samples. This procedure is repeated 1000 times, each time with a different set of test and training samples.

Finally, the overall performance is used to calculate the statistical mean and standard deviation of the measured accuracies, see results in Tab.II.

Classifier	CF	σ	D2	σ	SI	σ
SVM	85%	11%	68%	12%	83%	13%
MLP	86%	12%	60%	14%	76%	13%
kNN	83%	11%	65%	13%	83%	12%

TABLE II: Mean categorization accuracy results using the data set digitized with 3D scanner for Coarse Filter (CF), Distance Distribution (D2) and Spin Images (SI), together with respective standard deviation.

Especially the coarse filters and the spin images yield good results. Most problems occurred due to the confusion between similar categories, like apples and oranges or mugs and beakers, see Tab.III.

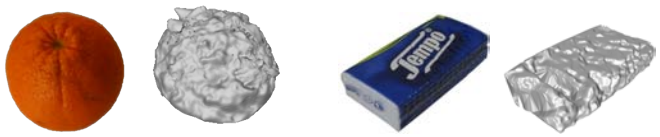
	Banana	Can	Apple	Orange	Beaker	Tissues	Mug	Bottle
Banana	70%	–	–	–	–	–	–	10%
Can	–	30%	–	–	30%	50%	–	–
Apple	–	–	100%	70%	–	–	–	–
Orange	–	–	–	30%	–	–	–	–
Beaker	–	70%	–	–	50%	50%	10%	–
Tissues	–	–	–	–	–	–	–	–
Mug	–	–	–	–	20%	–	90%	–
Bottle	30%	–	–	–	–	–	–	90%

TABLE III: Averaged confusion matrix of 10 SVM classifiers using distance distributions as features. Columns represent true labels, rows represent estimated categories by classifiers.

B. Noisy Data

In order to measure the influence of noise on the chosen features, another test was performed where a large amount of noise was added to the objects in the artificial data set. From the original point sets, 500 points were chosen as the basis for the new object. These points were then superimposed with 5 mm to 10 mm of evenly distributed noise. Even more, 3% of the points were chosen as outliers and their positions were changed by 5 cm to 10 cm, see Fig.10-b.

Subsequently, classifiers were trained on the (noise-free) training set, and evaluated on the test set with the added noise, see Tab.IV. The deterioration in accuracy can be attributed to two causes. For one, the features for noisy objects and objects without noise are different. Although a common assumption is that it is generally preferable to use a training set without noise, in this case the same experiment performed with the training data also being noisy achieved higher accuracy (the difference being 5% – 10%). Also, the introduced noise decreases the inter-class variance at the decision borders. The classifiers where unable to differentiate for example between the *orange* and *apple* categories and



a) b) c) d)

Fig. 11: Visual model reconstruction. a) Test object orange. b) Reconstructed test model orange. c) Test object tissue package. d) Reconstructed test model tissue package.

also the *beverage can* and *beaker* category became nearly indistinguishable.

Classifier	Coarse Filters	D2	Spin Images
SVM	61%	61%	63%
MLP	57%	59%	53%
kNN	59%	56%	57%

TABLE IV: Accuracy results for classifiers trained on noise-free training data set and tested on data set with artificial noise.

C. Partially Occluded Data

Experiments were also performed on artificially occluded objects by using only parts of the model, see Fig.10-c. Each object was used to create three occluded variants by cutting off a random part of the object. This was done by selecting a random plane intersecting the object and taking only the points on one side of the plane. It was ensured that the resulting object had a length of 30% to 70% along the chosen plane's normal compared to the original object, see Tab.V.

Classifier	Coarse Filters	D2	Spin Images
SVM	53%	42%	75%
MLP	60%	68%	61%
kNN	55%	56%	70%

TABLE V: Accuracy results for classifiers trained on data set without occlusion and tested on occluded data set.

D. Real Data

In order to evaluate the categorization approach with the humanoid robot, classifiers were trained on the training set with the scanned objects and applied to the models that were reconstructed by the humanoid robot. The test set comprised an apple, two bananas, a beaker, a beverage can, a bottle, a tissue package and an orange. Each object was captured from several views, and for each view a 3D point cloud was calculated which were then fused and used to create a 3D polygon model, see Fig.11. The trained classifiers were then applied to these 8 models, see results in Tab.VI.

Classifier	Coarse Filters	D2	Spin Images
SVM	100%	35%	50%
MLP	88%	35%	61%
kNN	70%	34%	51%

TABLE VI: Accuracy results for classifiers trained on (scanned) training set and evaluated on models reconstructed by the humanoid robot.

Surprisingly, the best results were achieved by the coarse filters, while spin images and distance distributions delivered less reliable results.

Classifier	Coarse Filters	D2	Spin Images
SVM	$_{\text{RBF}}\, C = 1, \gamma = 1$	$_{\text{RBF}}\, C = 1, \gamma = 1$	$_{\text{Linear}}\, C = 1$
MLP	64 Neurons	32 Neurons	20 Neurons
kNN	k=1	k=1	k=3

TABLE VII: Estimated classifier parameters for the different features.

E. Parameters

Proper parameters for the different feature vectors were empirically chosen. The histogram size for the distance distribution was set to 512, with the largest histogram bin corresponding to 3σ, where σ represents the mean distance.

The kernel density estimation was performed with an Epanechnikov kernel and the bandwidth was set to 0.1σ. For the histogram generation, 100,000 surface points were sampled and 100,000 point-pair distances were calculated.

The spin image size was set to 10, with bin size 6 mm (resulting in support distance 6 cm). Support angle was set to 120°. The spin image stack for training was constructed by taking the 16 k-means cluster centers calculated from 1000 random spin images. Categorization was performed on 100 random spin images using majority voting.

For the coarse filters, the best results were achieved by using the bounding box dimensions, bounding sphere radius, volume, convex hull area and convex hull volume as feature vector components. The estimated classifier parameters calculated by the grid search are presented in Tab.VII.

V. Conclusion

Shape-based object categorization is a challenging problem, especially when one is restricted to the limited visual sensing capabilities of a humanoid robot.

Up to now, no pure vision-based systems exist that were capable to generalize objects using their 3D shape. The difficulties of this task include varying lighting conditions, unfavorable object surfaces, context and self-occlusions, which results in incomplete or noisy reconstructions.

By exploiting the environmental knowledge while fusing several HDR stereo views from different view-points and applying robust reconstruction techniques, the humanoid robot is able to acquire sufficiently detailed 3D models of small objects in real application conditions such as difficult surfaces and unfavorable lighting.

The careful selection and proper transferring of the coarse filter approaches from 3D shape-retrieval to the object categorization task enabled the categorization of unknown objects by generalizing from known digitized samples. This promising results corroborate that model-based visual object categorization will enable humanoid robots to deal with unknown objects, consequently more general situations in real application scenarios.

VI. Acknowledgments

This work was partially conducted within the EU Cognitive Systems project GRASP (FP7-215821) funded by the European Commission and the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft).

REFERENCES

[1] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[2] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2002.

[3] Rosch E. Family Resemblances: Studies in the Internal Structure of Categories. *Cognitive Psychology*, 7(4):573–605, 1975.

[4] P.M. Roth and M. Winter. Survey of appearance-based methods for object recognition. *Inst. for Computer Graphics and Vision, Graz University of Technology, Austria, Tech. Rep. ICG-TR-01/08*, 2008.

[5] P. Azad, T. Asfour, and R. Dillmann. Combining appearance-based and model-based methods for real-time object recognition and 6d localization. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 5339–5344, 2006.

[6] K. Okada, M. Kojima, S. Tokutsu, T. Maki, Y. Mori, and M. Inaba. Multi-cue 3D object recognition in knowledge-based vision-guided humanoid robot system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3217–3222, 2007.

[7] M. Ulrich, C. Wiedemann, and C. Steger. Cad-based recognition of 3d objects in monocular images. In *International Conference on Robotics and Automation*, pages 1191–1198, 2009.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, volume 1, page 22. Citeseer, 2004.

[9] R. Fergus, P. Perona, A. Zisserman, et al. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2. Citeseer, 2003.

[10] R.B. Rusu, A. Holzbach, M. Beetz, and G. Bradski. Detecting and Segmenting Objects for Mobile Manipulation. In *Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan*, 2009.

[11] Radu Bogdan Rusu, Michael Beetz, Andreas Holzbach, Rosen Diankov, and Gary Bradski. Perception for mobile manipulation and grasping using active stereo. In *Humanoid Robots, 2009 9th IEEE-RAS International Conference on*, Paris, 12/2009 2009.

[12] A. Johnson. Spin-images: a representation for 3-D surface matching. *Robotics Institute. Pittsburgh, Pennsylvania: Carnegie Mellon University*, 1998.

[13] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3 d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[14] O. Stasse, S. Dupitier, and K. Yokoi. 3d object recognition using spin-images for a humanoid stereoscopic vision system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Beijing, China*, pages 2955–2960, 2006.

[15] N. Brusco, M. Andreetto, A. Giorgi, and G.M. Cortelazzo. 3D registration by textured spin-images. In *3D Digital Imaging and Modeling*, pages 262–269, 2005.

[16] P.A. De Alarcón, A.D. Pascual-Montano, and J.M. Carazo. Spin images and neural networks for efficient content-based retrieval in 3d object databases. *Lecture notes in computer science*, pages 225–234, 2002.

[17] D. Huber, A. Kapuria, R. Donamukkala, and M. Hebert. Parts-based 3d object classification. In *IEEE Computer Society Conference On Computer Vision and Pattern Recognition*, volume 2. IEEE Computer Society; 1999, 2004.

[18] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi. Humanoid robot HRP-2. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1083–1090. Citeseer, 2004.

[19] AS Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1584–1601, 2006.

[20] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, page 164. Eurographics Association, 2003.

[21] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, 2002.

[22] J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, and R. MacLeod. Coarse filters for shape matching. *IEEE Computer Graphics and Applications*, pages 65–74, 2002.

[23] M. Ben-Chen and C. Gotsman. Characterizing shape using conformal factors. In *Proceedings of Eurographics Workshop on Shape Retrieval*. Citeseer, 2008.

[24] J. Bohg, C. Barck-Holst, K. Huebner, M. Ralph, B. Rasolzadeh, D. Song, and D. Kragic. Towards Grasp-Oriented Visual Perception for Humanoid Robots. *In International Journal of Humanoid Robotics, Special Issue on Active Vision of Humanoids*, 6(3):387–434, 2009.

[25] N. Amenta, S. Choi, and R.K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153, 2001.

[26] T.K. Dey and S. Goswami. Tight cocone: a water-tight surface reconstructor. *Journal of Computing and Information Science in Engineering*, 3:302, 2003.

[27] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. *Shape Modeling International, Genova, Italy*, 2004.

[28] The KIT ObjectModels Web Database. `http://i61p109.itec.uni-karlsruhe.de/ObjectModelsWebUI`, September 2010.

[29] D. Gonzalez-Aguirre, T. Asfour, and R. Dillmann. Eccentricity Edge-Graphs from HDR Images for Object Recognition by Humanoid Robots. In *Humanoid Robots, 2010 10th IEEE-RAS International Conference on*, 2010.

[30] Nicole Atzpadin, Peter Kauff, and Oliver Schreer. Stereo analysis by hybrid recursive matching for real-time immersive video conferencing. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3), März 2004.

[31] J.W.H. Tangelder and R.C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008.

[32] C. Staelin. Parameter selection for support vector machines. *Hewlett-Packard Company, Tech. Rep. HPL-2002-354R1*, 2003.

Towards Robust Shape-Based Depth Image to 3D Model Matching using Inter-View Similarities

Walter Wohlkinger and Markus Vincze

Abstract— **Object recognition and especially object class recognition is and will be a key capability in home-robotics when robots have to tackle manipulation tasks and grasp new objects in an appropriate way or just have to search for objects. The goal is to have a robot classify never before seen objects within a single view in a fast and robust manner. The classification task can be seen as a matching problem, finding the most appropriate 3D model and view to a depth image. We introduce single-view shape model based approach using RGB-D sensors and a novel matching procedure for depth image to 3D model matching and inherently for object categorization. Our state-of-the-art ensemble of classifiers reliably delivers accurate classification results while being able to calculate the features directly from the 3D points of the sensor, without any calculation of normals or generating a mesh from it. We furthermore introduce a semi-automatic, user-centered approach to utilize the Internet for acquiring the required training data.**

We present an approach to fast and robust object classification on depth image data which can be acquired with any sensor delivering 3D point cloud data such as laser range scanners, stereo systems and RGB-D sensors like the PrimeSense sensor used in our Experiments This paper covers model acquisition from the web and a novel matching method by exploiting inter-view similarities of 3D models for increased object classification performance. Experimental evaluation on two common databases and a new hereby introduced database of real-world objects in a table-scene context was successfully performed.

I. INTRODUCTION

For service robots to enter real-world home environments, they have to become more adaptive to cope with changing environments and transfer knowledge from one setting to another. One of the key elements for robots to fulfil meaningful tasks like object search and retrieval or object manipulation is object and object class recognition. Human-robot-interaction, robot localization and mapping, and robotic manipulation can greatly benefit from a vision system which is able to categorize even never seen before objects at first glance.

The domestic setting with its plethora of categories and their huge intraclass variety demands a great deal of generalization skill from a service robot. These categories are characterized by their shape ranging from low intraclass diversification of fruits and simple objects like bottles up to high intraclass variety of liquid containers and furniture. To aggravate the scenario even more, the environment or even the task detains the robot from building a full 3D representation of space and objects around him by restricting its movements – no space to acquire views from around the object – or by restricting the time – a search task may be too slow if the robot has to move around every object for inspection.

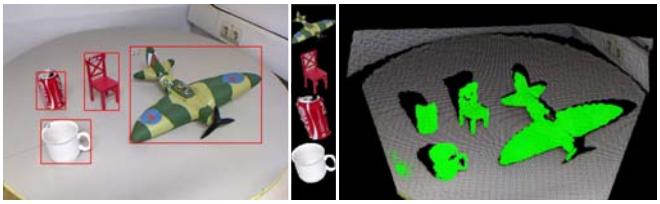


Fig. 1. Left: A recognized mug, a can, a toy-dining chair and a toy-plane on a turntable. Mid: The 2D segmentation of the objects. Right: The point cloud as produced by the PrimeSense sensor and the 3D segmentation of objects on the table plane.

Hence we propose a 2-fold strategy to tackle this problem. First, we use a single-view shape model based approach for range image to 3D model matching to give the system its required speed. Our methodology works directly on the 3D data without any need for time-consuming and sensor noise dependent operations such as normals calculation and mesh-generation from the point clouds. For increased matching performance we suggest to utilize inter-view similarity of the 3D models to discard false positives. This new matching scheme can be used with any global, affine invariant 3D descriptor to increase its performance.

Second, we grant a robot internet-access to use the information found on the internet to cope with the intraclass variation in categorization. By using 3D models from Google Warehouse[1] the problem of coping with a large intraclass variety is inherently solved, as the number of available models is proportional to the intraclass variety, reducing the problem from categorization to nearest-neighbour-matching. Now we only have to cope with scalability issues when matching against thousands of models, but this can be solved using approximate nearest neighbour and semantic hashing.

The classification is performed with multiple frames per second against a database of 3D models which can be generated and altered semi-automatically by a non-expert. Robust classification is achieved by choosing multiple complementary feature descriptors, choosing the appropriate similarity measure and combine the descriptors.

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

Vision4Robotics Group, Automation and Control Institute, Vienna University of Technology, Austria [ww,vm]@acin.tuwien.ac.at

[1] http://sketchup.google.com/3dwarehouse/

II. RELATED WORK

The focus of this work is to push the performance of solely 3D point based shape descriptors for range image to 3D model matching. Partial sensor view to 3D model matching was done by [2] using a dense SIFT based descriptor first introduced by [8] which is still the top performer in the SHREC shape retrieval contest of range images[2]. Extraction of such local features is computational expensive and the authors of [2] presented good results using multiple view with a movable sensor head. Moving around the object of interest, building a 3D model and then categorize the object using a humanoid robot was shown in [4]. They used Spin Images [5], D2 Shape Distribution [9] and geometric properties like bounding box and volume of the real-world sized 3D model, thus requiring acquisition of a specialized database with a structured light sensor. Using Spin Images was also done on 3D LIDAR point clouds by [3] and by [7] who also used 3D modesl from the web to match against. A global descriptor based on histograms of normals was introduced by [10] which delivered good results on container-like objects, but required calculation of normals.

III. METHODOLOGY

The categorization is based on matching depth images against a database of 3D models and a subsequent k-NN classifier. The stages of the system include the acquisition of the database, object segmentation and matching against the database.

A. Knowledge Acquisition & Model Preparation

The input into our model acquisition system is the name of the new object class, which can be entered by the user via voice or via keyboard. With this keyword we query the lexical database WordNet[3] to disambiguate the keyword by presenting the different meanings to the user to select the appropriate one. Knowing the correct meaning of the keyword, we now use the synonyms and hyponyms (words sharing a 'type-of' relationship with the keyword) provided by WordNet for the 3D model search on Google Warehouse[4]. After downloading of the models, the user selects one of the models as the reference model to enable a subsequent process of discarding wrong models from the database using a similarity criterion to the reference model. Having a semantic meaning and an index for the word in the hierarchy provided by WordNet enables further semantically meaningful manipulation applications like pouring something into a container-like object.

One way of matching range images to full 3D models is to see the problem as finding the appropriate view of the 3D model which can be achieved by formulating the problem as a partial-view to partial-view matching problem. To use the models from the web for depth image to depth image matching, we generate synthetic depth images by rendering the 3D models and sampling the z-buffer from 20 equally spaced views around the model using the vertices of a dodecahedron as done in the lightfield descriptor [12] and depicted in Figure 2. These 20 views are sufficiently dense for the type of descriptors used to interpolate between views. To discard details and therefore improve generalization of the models, we sample the models by rendering them in 150x150 pixel images which leads to around 5000 data points for the average model which fills the rendering window to 25%. Finally, for every one of the 20 views of the model the 3D descriptors are calculated and stored into the database. Using the appropriate distance measures, the best partial-view out of the 3D models can be found by comparing the descriptors calculated from the range image delivered by the sensor to all descriptors in the database. Classification is done using a k-NN classifier on the ranked results with k depending on the minimum numbers of models in the database for a class.

B. Matching with Inter-View Similarities

The basic idea is to match the range scan not only to one single view, but to several nearby views as nearby views share some similarities. These similarities depend on the type of descriptor used. For pure 2D shape descriptors, the silhouette will change with every variation of the viewpoint, but for pure 3D descriptors the change of the neighbouring views is less dramatic and can therefore be used to discard false positives. Figure **??** in the experimental evaluation depicts the improvement of using multiple view similarities.

C. Descriptors

The goal of classification is to find the correct class label for a given data cluster. This can also be seen as finding the most similar object to the query data and assigning the label of the most similar match. The use of multiple descriptors can lead to an increased recognition rate as descriptors don't perform equally on each category. When carefully chosen, the calculation overhead can be kept to a minimum by working on the same data and by sharing intermediate results.

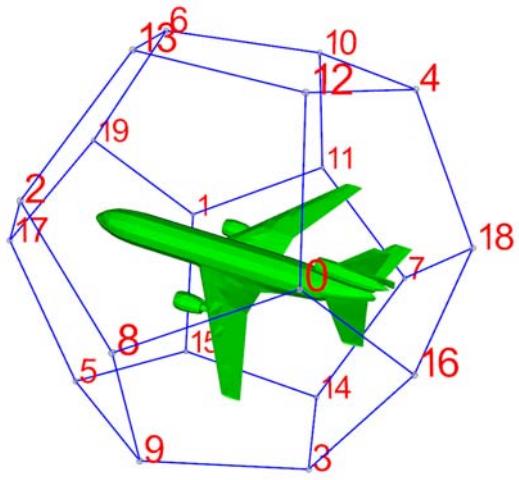


Fig. 2. A 3D model of the category "commercial plane" with the 20 viewpoints at the vertices of a dodecahedron.

[2]http://www.itl.nist.gov/iad/vug/sharp/contest/2010/RangeScans/
[3]http://wordnet.princeton.edu/
[4]http://sketchup.google.com/3dwarehouse/

We use the affine invariant shape distributions [9], moment invariants [11] and spherical harmonics [6] as descriptors as presented in the next sections.

1) D2 Shape Distribution: We use a multi-resolution version of the D2 shape distribution descriptor of [9] who introduced this descriptor for full 3D model matching. The advantage of this descriptor is that the histogram of distances between randomly sampled points can directly be calculated from the point cloud. To capture coarse structures and fine details, one has to find the best bin-size of the distance histogram. We avoid this by combining multiple bin resolutions into one histogram, as depicted in Figure 3. As our choice of distance measure we use the Taneja [1] similarity measure (Equation 1), which performed best across all classes in our evaluation of the similarity measures.

$$d_T = \sum_{i=1}^{d} \left(\frac{P_i + Q_i}{2} \right) \ln \left(\frac{P_i + Qi}{2\sqrt{P_i Q_i}} \right) \tag{1}$$

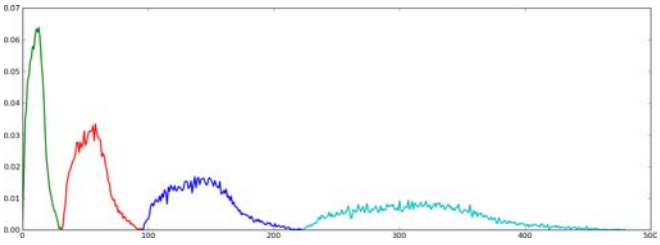


Fig. 3. The multi-resolution shape distribution histogram with bin size 32(green), 64(red), 128(blue) and 256(cyan) combined into a single descriptor.

2) Voxel based Spherical Harmonics: For this descriptor the point cloud is scaled to have a mean distance of 1 and voxelized into a cube with side length 64. The spherical harmonics for each of the 32 concentric spheres and for each of the 32 frequencies is precomputed and stored in a look-up-table. This enables this descriptor to be computed in a fixed amount of time. For the resulting 32 by 32 histogram we use K divergence [1] as the similarity measure given in Equation 2.

$$d_{Kdiv} = \sum_{i=1}^{d} P_i \ln \frac{2P_i}{P_i + Q_i} \tag{2}$$

3) Moment Invariants: For a coarse classification we use the moment invariants presented in [11] to improve our results. As the invariants are also calculated directly from the point cloud, there is only little overhead on calculating this additional descriptor. The invariants are stacked into a single vector and as the similarity measure of choice for this descriptor we decided on Wave Hedges [1] given in Equation 3.

$$d_W = \sum_{i=1}^{d} \frac{|P_i - Q_i|}{\max(P_i, Q_i)} \tag{3}$$

IV. EXPERIMENTAL EVALUATION

We demonstrate the increased performance separate on the descriptors with a sample query on the Princeton Shape Benchmark to clearly single out the advantage of using our proposed matching scheme. Figure 6 shows as range scan generated from rendering the 3D model and sampling the z-buffer. The descriptors are calculated from this point cloud and matched against 20 categories.

For evaluation of our approach we used three databases for evaluation of the different aspects of our approach. On each database we used as a measure of retrieval performance *First Tier (FT)*, *Second Tier (ST)*, *Nearest Neighbour (NN)* and the *precision recall (PR)* curve together with *Average Precision (AveP)* to provide a more detailed look into the performance of the system.

A. PSB: Synthetic Data

The Princeton Shape Benchmark (PSB) database is a common database for comparing 3D model retrieval with 1814 models in 92 categories. We used the database to test the scalability of our approach to a high number of categories.

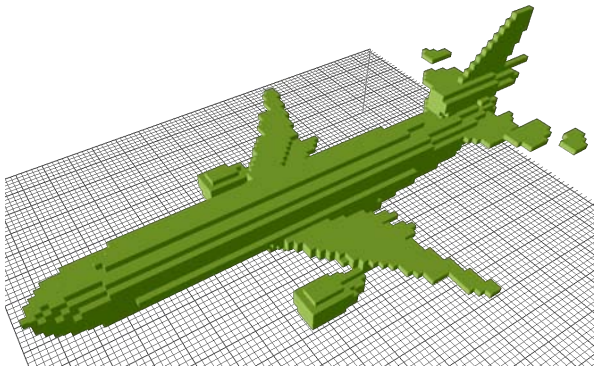


Fig. 4. The voxelized point cloud of a plane.

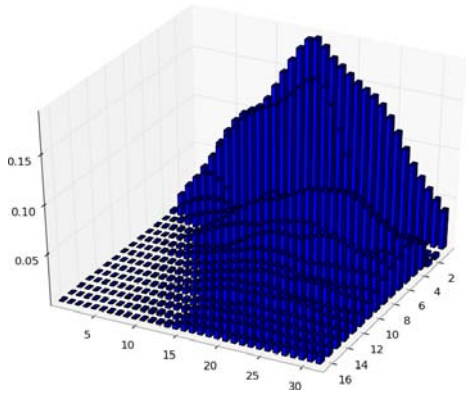
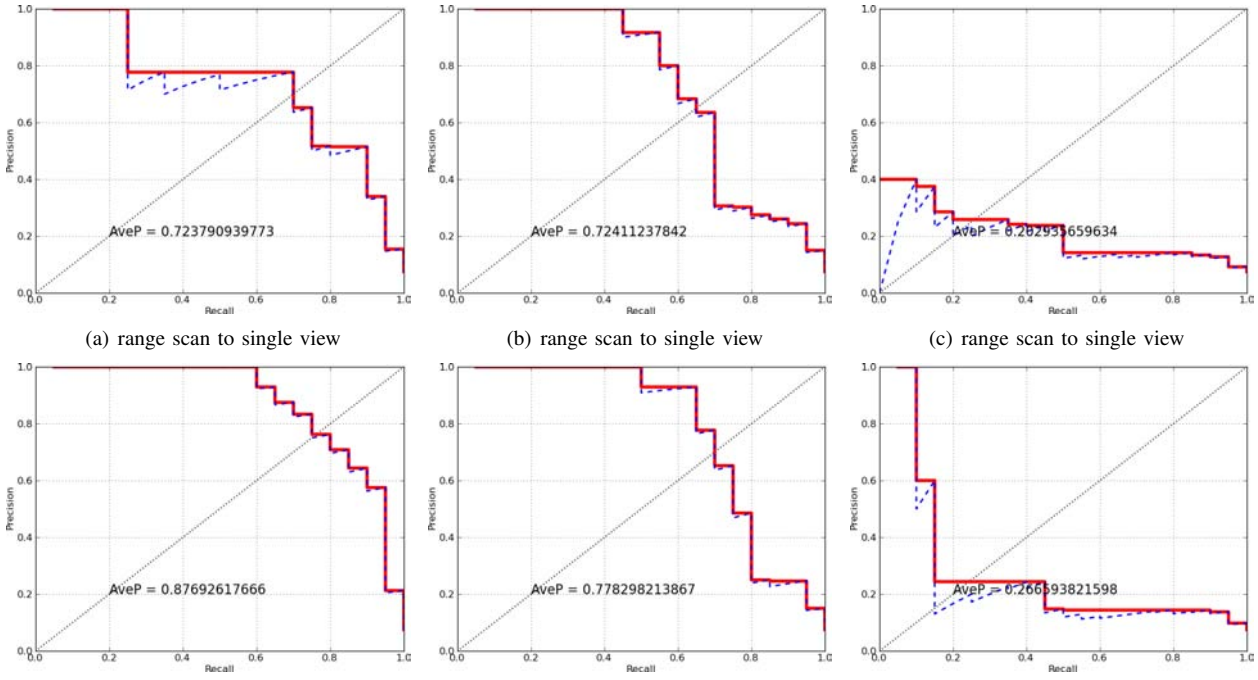


Fig. 5. The voxel based spherical harmonics descriptor [6] applied to partial view data.



(a) range scan to single view (b) range scan to single view (c) range scan to single view

(d) range scan to multiple views (e) range scan to multiple views (f) range scan to multiple views

Fig. 7. BlaBlaBla

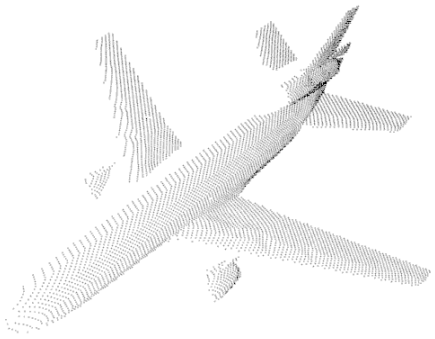


Fig. 6. A partial view of a commercial airplane to be matched against the database.

B. SHREC: Range Image Contest

The SHREC Range Image Matching benchmark is a annually held contest to retrieve the object classes from scanned objects on a table using a high resolution laser scanner.

C. CatDB: a RGB-D Database

We introduce a new database for testing object classification acquired with a RGB-D camera. The database provide tools for capturing scenes from a Kinect sensor, annotate the scenes and to replay selected scenes to ease testing.

V. CONCLUSION

In this paper we investigated the use of Web-learned models to detect object classes in depth images from actual scenes. The intention was to use the object class relation to derive grasp points fort he respective objects. We implemented a scheme to learn view-based 3D models given the Web data. This reference model can be used for matching with the depth data provided by a state-of-the-art RGB-D sensor such as the PrimeSense sensor. The results clearly indicate that the mixture of features used to describe the object models achieve high recognition rates. We further showed that with using multiple views at the matching stage the average precision can be considerably improved.

The advantage of this approach is that new object class models can be very efficiently learned from Web data and that matching is robust and fast using the depth images. Future work comprises the investigation of more and alternative features and a deeper analysis of the cases where pure matching of 3D data is misleading and should be complemented by adding appearance data to the object class models.

REFERENCES

[1] Sung-Hyuk Cha. Taxonomy of nominal type histogram distance measures. In *Proceedings of the American Conference on Applied Mathematics*, pages 325–330, Stevens Point, Wisconsin, USA, 2008. World Scientific and Engineering Academy and Society (WSEAS).

[2] C. Goldfeder, M. Ciocarlie, J. Peretzman, Hao Dang, and P.K. Allen. Data-driven grasping with partial sensor data. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1278 –1283, 2009.

[3] Aleksey Golovinskiy, Vladimir G. Kim, and Thomas Funkhouser. Shape-based recognition of 3d point clouds in urban environments. *ICCV*, 2009.

[4] D. Gonzalez-Aguirre, J. Hoch, S. Roehl, T. Asfour, E. Bayro-Corrochano, and R. Dillmann. Towards shape-based visual object categorization for humanoid robots. In *ICRA*, 2011.

[5] 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.

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

[7] Kevin Lai and Dieter Fox. Object detection in 3d point clouds using web data and domain adaptation. *International Journal of Robotics Research*, 2010.

[8] 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, 272009-oct.4 2009.

[9] 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.

[10] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2155 –2162, 2010.

[11] Firooz A. Sadjadi and Ernest L. Hall. Three-dimensional moment invariants. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-2(2):127 –136, 1980.

[12] Yu-Te Shen, Ding-Yun Chen, Xiao-Pei Tian, and Ming Ouhyoung. 3d model search engine based on lightfield descriptors. In *Eurographics*, 2003.

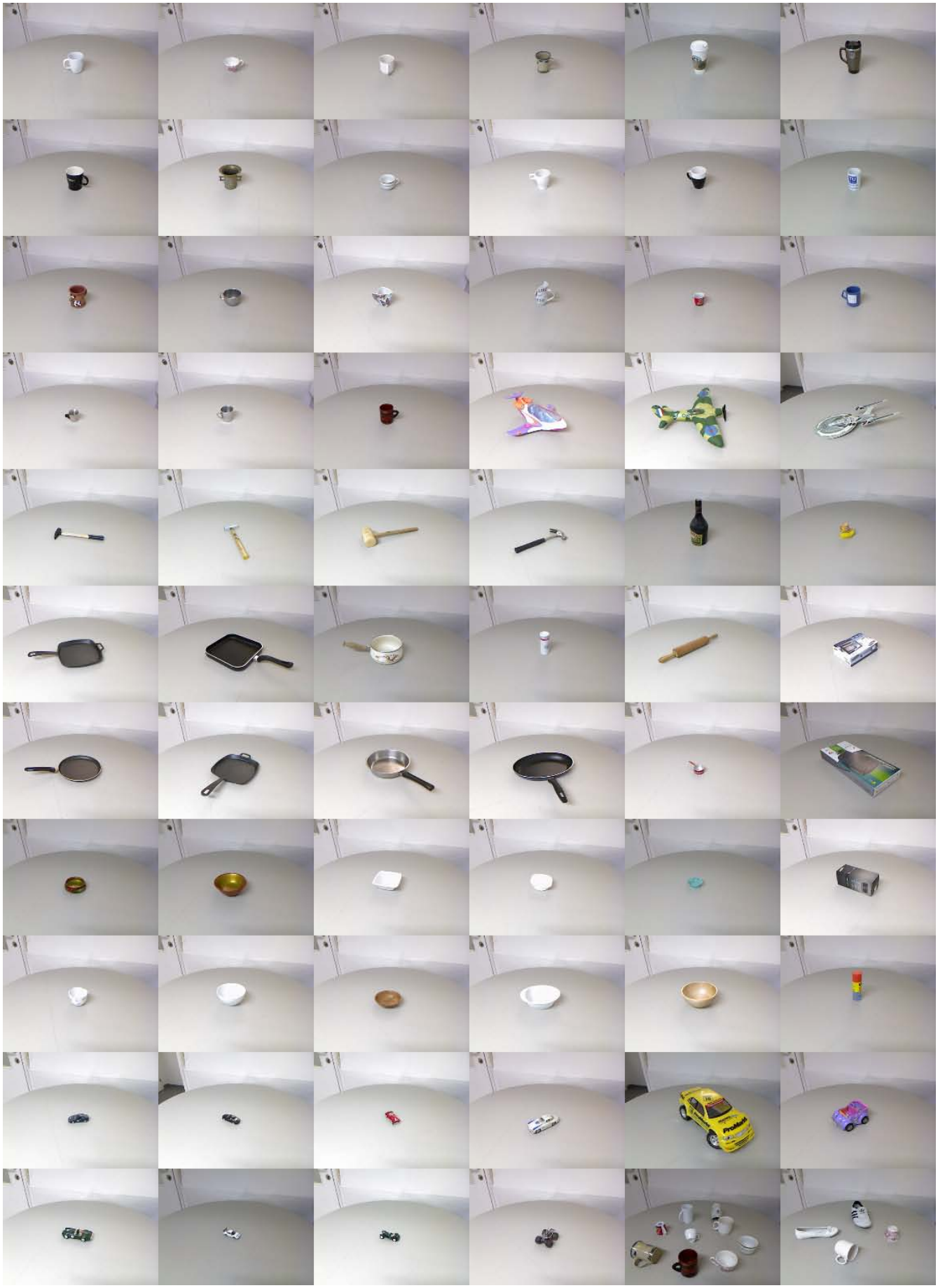


Fig. 8. The objects in the CatDB: Each object is captured by 16 views around the object and is then combined with other objects to create a more realistic scenery. The last two images depict how the mixed scenes look like.

Pose Alignment for 3D Models and Single View Stereo Point Clouds Based on Stable Planes

Anonymous 3DIMPVT submission

Paper ID 120

Abstract

This paper presents a novel method for alignment of geometrically similar 3D models. It is based on the prior that both models have at least one common tangent plane on which both can stand stably and when standing on it the models are partially aligned. To determine the final rotation around the stable plane's normal, needed for a complete alignment, we adapt an image alignment technique based on the log-polar transformation. Because the set of stable planes of a model is small enough, alignment is efficiently approached as a global optimization problem that finds the common stable plane providing the best alignment according to a similarity measure. As the method does not rely on any kind of global symmetry features, we show it can be used to register incomplete stereo point clouds of objects located on a stable plane (table, ground, etc.) with the corresponding similar 3D models. We evaluate the 3D-alignment method by comparing it to the well-known CPCA and show a significant improvement when aligning 120 *models belonging to* 12 *different classes.*

1. Introduction

Pose alignment has been extensively used to provide a canonical reference system for 3D model databases to reinforce 3D model retrieval [15] using pose-dependant 3D descriptors.

Not only computer graphic researchers are using 3D databases but also robotic and computer vision researchers have been adopting these sources of information to reinforce well-researched but still open issues, like object recognition/classification ([10], [9]), object grasping ([5], [4]), etc.

In these cases, pose alignment/registration is needed to provide the link between 3D models (represented in a local reference system and usually with an arbitrary alignment) and the real world objects. 3D pose estimation in real scenarios faces noisy and incomplete data due to sensor and algorithmic limitations and constrained view ranges which makes the task even more challenging. This is one of major drawbacks of the pose alignment methods proposed by the computer graphic researchers: the difficulty to apply them to align 3D models with incomplete sensed models.

Our final goal is to solve grasping of novel objects using a data-driven grasp approach. We reduce the problem of object grasping to finding the most geometrically similar object that we already know how to grasp and transfer the known grasp, similar to [5], [4]. Objects which are geometrically similar, will be grasped in a similar fashion. Object grasping is a complete pose dependant problem and therefore, in order to use valuable information provided by sources like the "Columbia Grasp Database" (CGDB [4]), pose registration between the models and the real scenario objects is required.

The contribution of this work is two-fold: we present a novel 3D alignment method superior to the well-known "Continous Principal Components Analysis" (CPCA [15]) and we show that its properties make it suitable to be used for alignment of real scenario objects with 3D models. The prior that objects stand on a stable plane is fullfilled in most of the situations and second, avoiding the use of global object properties like axial symmetries or moments allow us to work with incomplete sensed models.

The rest of the paper is structured is as follows: in Section 2, the most relevant related work in pose alignment is summarized. Section 3 presents our method and Section 4 shows how it can be adapted to be used with sensed data together with some preliminary results. Section 5, presents an evaluation of the method and we finally conclude with some future ideas.

2. Related work

Pose normalization has historically been performed by "Principal Component Analysis" (PCA), which is based on the computation of moments of 3D models. The principal axes given by PCA are used to align the models after translating the center of mass (CoM) to the origin of the coor-

dinate system. The most stable of all PCA-approaches is known to be the CPCA ([15]). Chaouch et al. [2] gives a detailed explanation for which object classes CPCA will succeed based on plane reflection symmetry analysis.

Kazhdan [7] approaches pose alignment as an optimization problem and reduces the computational effort using parametrization techniques which allow to optimize independently over smaller subspaces. Kazhdan [7] proposes to use the axial symmetry properties of the models to factorize the search space and shows that an efficient and optimal alignment can be found for axial symmetric models. The method computes a meaningful as possible approximation for other types of models.

To be able to align 3D models with incomplete sensed models, we must avoid using any methods that make use of symmetry properties as it will be impossible to compute them when dealing with incomplete data.

Regarding alignment of incomplete data, some researchers have proposed using correspondences between local features [6]. Such methods have two main difficulties: selecting salient local features that work across different types of objects and second, globally similar objects may not share enough local features for alignment.

Goldfeder et al. [5] align partial range scans with neighbor models by breaking alignment into a rough stage and a refinement stage, using the "Iterative Closest Point" [12] algorithm. The rough stage is the initial approximation used for ICP [12] which is known to need a good approximation in order to be successful. The major problem in their rough stage is to assume that the CoM of the sensed model, which is normally a partial view of the whole object, will be close to the CoM of the actual object.

We approach the alignment problem as a global optimization problem and reduce the search space by using stable planes instead of symmetry properties like [7]. By using stable planes, the method can be used to align objects in real scenarios because the plane (table, ground, etc.) where the actual object stands can be computed accurately.

3. Pose alignment for 3D models

Figure 1 depicts the flow of the proposed method. To align a query model with a target model, the algorithm selects a stable plane from each model (1) and rotates the objects so that the plane's normals coincide. As we are constantly working with the models in our database, the stable planes are computed for all the models in a pre-processing step. Once the objects stand on the plane, the models are sampled and projected on the stable plane (2). The projected histograms are then aligned (rotated, scaled) using 2D shape alignment (3). The models similarity is evaluated (4), the algorithm returns to (1) and selects the next plane combination.

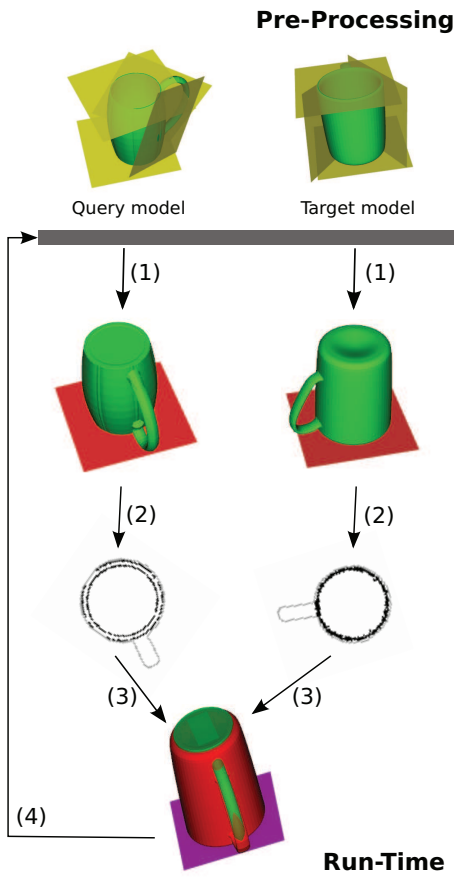


Figure 1. Algorithm flow when aligning two models. The yellow planes represent the stable planes of each cup. After (1), if a good combination of planes is selected, the models stand on a common plane and partially aligned. (2) projects the models on the plane and (3) rotates and scales the histograms to obtain a maximal correlation. The computed rotation and scale are used to transform the models which are completely aligned after (3).

3.1. Pre-Processing: Stable planes computation

As noticed in [3], the stable planes of a model are a subset of the tangent planes enclosing a model which are the planar faces of the convex hull. In our implementation, we perform a hierarchical clustering [1] to group the triangles of the convex hull in planar faces. The final clusters represent the tangent planes Π.

Consider $\pi \in \Pi$. The model is rotated in such a way that the normal of π matches the y-axis of the world coordinates and translated to stand on π. Let $cH(s)$ be the 2D convex hull of the points supporting the plane, $cH(proj)$ the 2D convex hull of the whole projected model onto π and $A(pol)$ a function returning the area of a polygon pol. π is a stable plane if the projection of the center of mass of the object lies inside $cH(s)$.

In our case, different to [3], we are not just interested in the upright orientation base. We would also like to align models when they are lying on different orientations. Nevertheless, being able to sort the stable planes, will allow us to check first those stable planes with similar properties with respect to the model. Therefore, we sort the stable planes using the following equation:

$$p(\pi) = A_r * (d_1 + d_2) \tag{1}$$

where

$$A_r = A(cH(s))/A(cH(proj)) \tag{2}$$

captures the static stability of π and d_1,d_2 represent the *coincidence distance* and *collinearity distance* approximating symmetry characteristics of π as described in [3].

Although we consider this step as a pre-processing step, if an unknown model needs to be aligned, the stable planes can be efficiently computed as it is just needed to compute the convex hull of the model, group triangles in planar faces and check for each planar face the condition stated before.

3.2. Aligning the models

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ represent the models to be aligned after translating their CoM to the origin. Being Π_1 and Π_2 respectively the stable planes of $\mathcal{M}_1$ and $\mathcal{M}_2$, the optimization problem consists in finding the combination (Π_1^i, Π_2^j), scale factor **s** and rotation **r** around the plane's normal providing the best alignment. Hence, we have $|\Pi_1| * |\Pi_2|$ possible combinations and the best alignment is found by the following equation:

$$(i, j, r, s) = \underset{\Pi_1, \Pi_2}{\operatorname{argmin}} \mathcal{D}\left(\mathcal{A}_{r,s}\left(\mathcal{R}(\mathcal{M}_1, \Pi_1^i), \mathcal{R}(\mathcal{M}_2, \Pi_2^j)\right)\right) \tag{3}$$

The function $\mathcal{R}(\mathcal{M}, \pi)$ rotates the model so that the normal of π coincides with the canonical y-axis and translate $\mathcal{M}$ along the y-axis in order to stand on π. Define now $\mathcal{M}' = \mathcal{R}(\mathcal{M}, \pi)$. $\mathcal{M}'_1$ and $\mathcal{M}'_2$ are now models laying on the same plane for a combination of (i, j). Assuming, Π_1^i and Π_2^j would provide a good alignment base for $\mathcal{M}_1$ and $\mathcal{M}_2$, we still need to get rid of the last degree of freedom around the normal of the plane. Indeed, we need to find the rotation **r** around the plane normal, that provides an optimal alignment for $\mathcal{M}_1$ and $\mathcal{M}_2$. For this porpouse, we could compute the principal components (PCA) of the projected points on the plane and align them. As Figure 2 shows, the result will be satisfactory when both sets of projected points present a dominant direction but will fail otherwise. Instead of using PCA, $\mathcal{A}_{r,s}\left(\mathcal{M}'_1, \mathcal{M}'_2\right)$ computes the rotation **r** using a 2D shape alignment technique (see section 3.2.1) which moreover provides a scale factor **s** used to scale the models. After the 2D shape alignment, our initial models have been transformed to its corresponding translated, scaled and rotated versions: $\mathcal{M}''_1$,$\mathcal{M}''_2$.

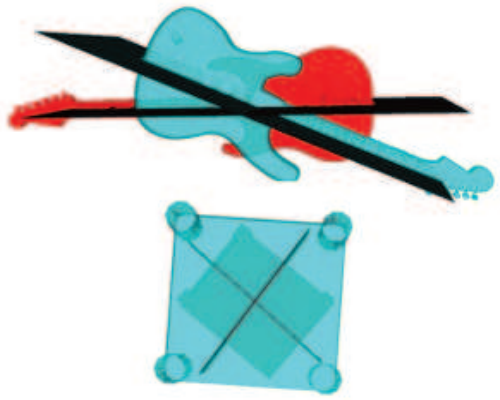


Figure 2. Top row: 2D PCA will align the models correctly. Bottom row: 2D PCA will fail. The dark planes represent the principal components of the models.

The final point of equation (3) is the function $\mathcal{D}$ which is a dissimilarity function:

$$d = \mathcal{D}(\mathcal{M}''_1, \mathcal{M}''_2) \in \Re, d > 0 \tag{4}$$

When pose alignment is approached as an optimization problem, a metric is needed to quantify how similar or different the models $\mathcal{M}''_1$ and $\mathcal{M}''_2$ are at the current optimization step. Therefore, the dissimilarity measure must be pose-dependant. Hence, $\mathcal{D}$ compares two models by obtaining 3 different pairs of views from $\mathcal{M}_1$ and $\mathcal{M}_2$ and compute the contour distance for each pair of views using the well-known distance transform. The views are obtained by placing a virtual camera outside the model on the x, y, z axes and configured to create a parallel projection of the model. The view model is similar to the cross-section, floor plan and elevation model used in CAD modelling.

3.2.1 2D shape alignment

The image registration technique presented in [16] is able to compute the rotation and scale needed for an image patch in order to match the original image. Transforming to the log-polar space enables to find the needed rotation and scale by performing ordinary cross-correlation over the transformed images. Our problem is slightly different as the shapes we try to register do not belong to the same object. Even so, the cross-correlation will find the rotation and scale where both shapes match at most and hence, the best alignment.

We use the method in [16] (without a final affine registration step) and feed it with projection histogram images $\mathcal{HF}_1$, $\mathcal{HF}_2$ of the models onto the stable plane. The 3D models are uniformly sampled (using the technique presented by Osada et al. [11]) and the sampled points projected on the stable plane are used to build a projection his-

togram ($\mathcal{H}$). Defining σ and $\bar{h}$ as the average value and standard deviation of $\mathcal{H}$, $\mathcal{H}_e$ as the result of the edge detector on the binarized $\mathcal{H}$ and $\mathcal{H}_i = \mathcal{H} - \mathcal{H}_e$, the final histogram $\mathcal{HF}$ is built as follows:

$$\mathcal{HF} = \{\mathcal{H}_i, \mathcal{H}_i(u,v) >= \bar{h} + 2\sigma\} \cup \mathcal{H}_e \qquad (5)$$

which provides a histogram composed of the edge information of the projected model ($\mathcal{H}_e$) and the most distinctive parts inside of that profile in the plane's normal direction (back seat surface on chairs, legs on tables, etc.). $\mathcal{HF}$ is binarized and used as input for the log-polar cross-correlation. Figure 3 shows the importance of using the salient information along the plane's normal.

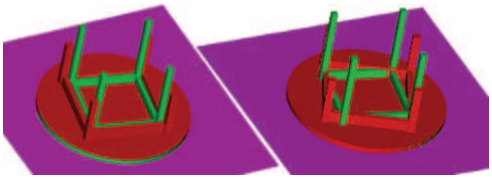


Figure 3. By using the projected histogram, the cross-correlation takes into account the salient information on the plane's normal direction and correctly aligns the legs of the table (left part of the image). The right part shows the alignment when only the edges of the projection are used. It can be seen that a random orientation is choosen for the legs.

The method provides a scale factor (**s**) in the XZ plane that can directly be used to scale the whole model isotropically. We avoid anisotropically scaling as it will lead to deformations on the structure of the object. Using directly (**s**) to scale the model discards all the scale information given by the model along the plane's normal direction. Therefore, a scale factor $\mathbf{s}'$ is computed along that direction (see [15]). The models are scaled independently using (**s**) and ($\mathbf{s}'$) and the scale factor giving the smallest dissimilarity d is choosen.

3.2.2 Computational requirements

The complexity of the method is $O(m \times n \times d \times s)$, where m and n are the number of stable planes in the models we would like to align, d is the size of the projected histogram (see Eq. (5)) and s is the scale range we want to check when computing cross-correlation. The histogram size d is 256 and s ranges in $[-5, 5]$ in the log-polar space meaning that the query model may be 20% smaller or bigger than the target model.

Because the stable planes are sorted similarly it is not necessary to check all the possible combinations. According to our experiments, if we set the upper bounds for the iterators i,j (see Eq. (3)) to 3 and 5 respectively, we still have a high probability that any of these 15 combinations will contain a common stable plane which provides a good alignment base. If one of the planes is fixed, as when we want to align a real object in a scene, we set $i = 1$ (ground-table plane) and probably all the possible combinations for $\mathcal{M}_2$, so that, $j = |\Pi_2|$.

4. Aligning sensed models with 3D models

In this section, we present how the method can be adapted to align sensed models, reconstructed from a single stereo view, with the corresponding similar 3D model. In this situation, the reconstruction of the sensed model will be incomplete as some parts of the object are not seen from the camera point of view and some visible parts will be missing or noisy. Due to the unseen parts of the object, the CoM of the reconstruction projected on the plane is shifted from its actual position. Therefore, we cannot longer use the centers of mass to center the models as usually done for 3D models alignment. Hence, instead of having two degrees of freedom (rotation around the plane normal, scale), we have now two more: the actual center of mass coordinates of the sensed model on the plane.

The multiscale approach of the log-polar registration, also presented in [16], is able to determine the translation between two images (histograms in our case) that we use to center the models. At a given resolution, the projected histogram of the sensed model is translated to different positions, the log-polar transformation is computed and cross-correlated with the log-polar transformed histogram of the 3D model to determine rotation and scale. The parameters giving the highest correlation are used as an approximation for the next level. Additionally, the log-polar registration can cope with model inclompeteness and it is robust enough to succeed with noisy data to a certain degree.

Regarding model incompleteness, although cross-correlation is a global measure, it is still able to determine the parameter location where the projections match at most with each other even when one of the projections is incomplete. One can think about cross-correlation as a local features approach with the advantages that the features do not need to be explicitly specified and that all the information available is used.

Because the scale between real objects and the 3D models database could be bigger than the scale range accepted by the log-polar registration, a preliminary step to approximate the scale of the models is needed. Scaling the models so that the dimensions along the plane normal match (like when computing $\mathbf{s}'$) ensures that the scale factor will be small enough and therefore, inside the range checked by the cross-correlation.

Figure 4 shows models reconstructed from stereo data aligned with the corresponding 3D models on the database. The reconstructed models where obtained from virtual stereo cameras by taking a view of the model lying on a

stable plane. As stereo algorithms need textured regions to estimate depth, each triangle on the model is painted using a random color.

Please note, that when a higher precision is needed, the alignment given by our method can be used as an accurate starting approximation for ICP [12] as done by other researchers [5].

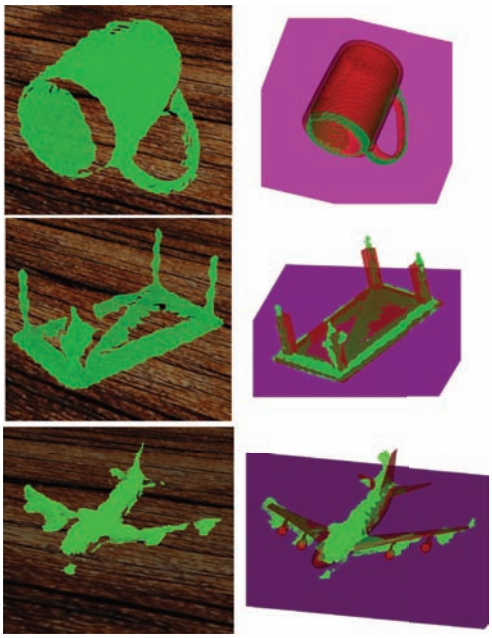


Figure 4. Left: 3D reconstruction from stereo. Right: Stereo point cloud (green) aligned with the most similar 3D model in our database (red) using our method. Best viewed in color.

5. Evaluation

To evaluate the 3D pose alignment algorithm we have randomly selected 120 models belonging to 12 classes included in the Princeton Shape Benchmark [14] (PSB). Given a query model $\mathcal{M}_i$, the target model ($\mathcal{M}_j, j \neq i$) is selected by finding the closest geometrical (according to Spherical Harmonics Descriptors [8]) model belonging to the same class than $\mathcal{M}_i$. PSB contains a tree-like classification used to find the possible target candidates belonging to a given class.

The performance is evaluated (Table 1) by using the dissimilarity measure we introduced before and computing it for each pair of aligned models after being aligned with CPCA [15] and the presented method. It can be seen that our method is superior than CPCA.

In Table 2 the evaluation is performed using the Spherical Extent Descriptor [13]. CPCA performs slightly better than our method for categories on which the principal components are strongly defined (tool, human, vehicle and blade). Helicopters are better aligned with CPCA for two reasons: in some cases, our prior is not fullfilled as the stable plane providing the alignment base is not found due to a non uniform mass distribution of the actual helicopter (3D models are assumed to have a uniform mass distribution) and in other cases, the 2D shape alignment gets a higher correlation when aligning the rotor blades (mobile part of the model) instead of the body. Even so, as the method presented outperforms CPCA for other classes (see Figure 5), the average performance of our method is slightly better than CPCA when using the Spherical Extent Descriptor for comparison.

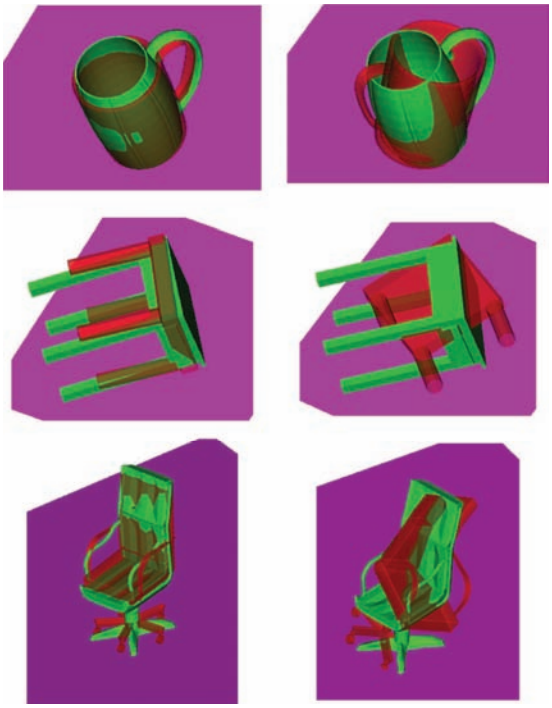


Figure 5. Left: 3D models aligned using our method. Right: Same models aligned using CPCA. Stable plane is also shown. Best viewed in color.

6. Conclusions and future work

We have presented a novel pose alignment approach that can be applied to a wide class of shapes. It is able to efficiently align 3D objects with other geometrically similar objects by using the prior that similar objects will share one or more stable planes which provide an alignment base.

Using this prior instead of global properties of the objects, allowed us to show that the method can be adapted to align point clouds reconstructed from a single stereo view with the corresponding 3D model.

Nevertheless, we believe some improvements are still

	Avg			Max	
	CPCA	Ours	Diff	CPCA	Ours
animal	1.98	1.66	+0.33	2.97	2.83
tool	0.75	0.80	-0.04	1.50	1.51
airplane	1.09	0.95	+0.13	2.57	1.76
helicopter	1.76	1.48	+0.28	3.29	2.18
vehicle	1.43	1.31	+0.12	3.14	2.82
human	0.74	0.70	+0.05	1.28	1.19
liquid c.	2.26	1.50	+0.77	5.71	3.53
gun	1.55	1.23	+0.32	2.57	2.01
seat	2.37	1.64	+0.74	4.50	4.95
blade	0.73	0.84	-0.11	1.75	2.05
shelves	1.71	1.47	+0.24	3.52	3.23
table	2.30	1.70	+0.61	5.50	2.69
All	1.56	**1.28**		5.71	**4.95**

Table 1. Results aligning 120 models with the proposed method and CPCA. To generate the results for our method we have used the first three stable planes for $\mathcal{M}_i$ and the first five for $\mathcal{M}_j$. The metric used for comparison is the dissimilarity measure we introduced before.

	Avg			Max	
	CPCA	Ours	Diff	CPCA	Ours
animal	1.30	1.20	+0.10	1.93	1.65
tool	0.55	0.62	-0.07	1.17	1.49
airplane	1.03	0.95	+0.08	2.08	1.66
helicopter	0.84	1.03	-0.19	1.22	1.76
vehicle	1.36	1.40	-0.04	3.69	3.63
human	0.66	0.70	-0.04	1.15	1.12
liquid c.	1.97	1.59	+0.38	4.61	3.64
gun	0.92	0.85	+0.07	2.00	1.29
seat	2.44	2.21	+0.23	3.10	3.03
blade	0.30	0.48	-0.18	0.46	1.31
shelves	1.90	1.91	-0.01	3.09	3.24
table	2.46	2.12	+0.34	4.48	2.84
All	1.32	**1.26**		4.61	**3.64**

Table 2. Results aligning 120 models with the proposed method and CPCA. To generate the results for our method we have used the first three stable planes for $\mathcal{M}_i$ and the first five for $\mathcal{M}_j$. The metric used for comparison is the L_2-difference between the spherical extent descriptors of the models.

possible when computing the final rotation around the alignment base by improving the 2D shape alignment method or even by correlating 3D descriptors. In any case, the method used should fullfill the requirements, stated during the paper, for alignment of sensed models. Besides, we will be implementing the algorithm in our robotic arm to test the grasping approach (includes the proposed alignment) in real robotic scenarios.

References

[1] M. Attene, B. Falcidieno, and M. Spagnuolo. M.: Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22:181–193, 2006. 2

[2] M. Chaouch and A. Verroust-Blondet. Alignment of 3d models. *Graph. Models*, 71(2):63–76, 2009. 2

[3] H. Fu, D. Cohen-or, G. Dror, and A. Sheffer. Upright orientation of man-made objects. *ACM Trans. Graphics*, pages 1–7, 2008. 2, 3

[4] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen. The columbia grasp database. In *IEEE Intl. Conf. on Robotics and Automation*, 2009. 1

[5] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen. Data-driven grasping with partial sensor data. 1, 2, 5

[6] D. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(1):637–650, July 2003. 2

[7] M. Kazhdan. An approximate and efficient method for optimal rotation alignment of 3d models, 2006. 2

[8] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, June 2003. 5

[9] K. Lai and D. Fox. 3d laser scan classification using web data and domain adaptation. 1

[10] A. Nuchter, H. Surmann, and J. Hertzberg. Automatic classification of objects in 3d laser range scans. In *In Proc. 8th Conf. on Intelligent Autonomous Systems*, pages 963–970. IOS Press, 2004. 1

[11] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21:807–832, 2002. 3

[12] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *International Conference on 3-D Digital Imaging And Modeling*, 2001. 2, 5

[13] D. Saupe and D. V. Vranic. 3d model retrieval with spherical harmonics and moments. In *DAGM*, pages 392–397. Springer-Verlag, 2001. 5

[14] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *In Shape Modeling International*, pages 167–178, 2004. 5

[15] D. V. Vranic, D. Saupe, and J. Richter. Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. In *IEEE MMSP 2001*, pages 293–298, 2001. 1, 2, 4, 5

[16] G. Wolberg and S. Zokai. Robust image registration using log-polar transform. In *In Proc. IEEE Int. Conf. image processing*, pages 493–496, 2000. 3, 4

Affordance based Part Recognition enabled Visual Cognitive Engine for Grasping and Manipulation

Karthik Mahesh Varadarajan, Markus Vincze

***Abstract*—Affordances (Unit utility, functional and topological relationships) and semantic scene understanding are key to building a generic, scalable and cognitive architecture for visual perception. 'Affordance based object recognition' or recognition based on affordance features is an important step in this regard. In this paper, we extend the scope of affordance features to define '*Conceptual Equivalence Classes*' and to recognize these classes leading to a Visual Cognitive Engine. This schema is analogous to building a scalable unit (part/ part assembly/ object) recognition system. This system has been built with the target of enabling a robotic arm to grasp a-priori unseen objects in a cluttered scene using global knowledge. The Visual Cognitive Engine uses inputs from multiple ontology bases – ConceptNet (for unit concept definitions), WordNet/WebKB (for textual unit definitions), Otto Bock Human Grasping database (for grasp affordances), ImageNet and other resources (for visual unit definitions), together with the Part Functional Affordance Schema, developed in this paper, to establish semantic unit affordances. Recognition of parts or part assemblies based on affordances, using the Visual Cognitive Engine enables knowledge of affordance and interaction modes for the entire object. This leads to a goal-directed object recognition and manipulation system that can perform implicit cognitive tasks such as substitute a cup for a mug, bottle, jug, pitcher, pilsner, beaker, chalice, goblet or any other unlabeled object, but with a physical part affording the ability to hold liquid and a part affording grasping, given the goal of 'bringing an empty *cup*' and no cups are available in the work environment. The performance of such a system is superior to traditional view-point based object recognition and manipulation systems.**

Keywords-Affordance, Recognition by Components, Part affordances, Grasping, Cognitive Object Recognition

I. INTRODUCTION

Semantic scene understanding is key to building a holistic visual object recognition and manipulation system for robots. The Semantic Robot Visual Challenge (SRVC) [28] provides a platform for research in this direction. SRVC is a research competition that is designed to utilize automatic acquisition of knowledge from large unstructured databases of images. Fully autonomous robots utilize learn visual models of queried objects from the web in order to identify the objects in the robot's cameras. Other projects such as RoboEarth [29] also utilize the web as a knowledge resource for model based object recognition and manipulation. Embodiment based scene understanding using RACER logical ontology base and proto-object definitions have been studied in [10].

On a more generic level, besides robotics, Semantic Web based knowledge acquisition systems have been typically defined using Web Ontology Languages (OWL), that are characterized by formal semantics and RDF/XML-based serializations. Extensions to OWL have been used in semantic editors such as Protégé and semantic reasoners and ontology bases such as Pellet, RacerPro, FaCT++, HermiT, etc. In the area of semantic text parsing and knowledge management, a number of frameworks such as Framenet, Lexical Markup Framework (LMF), UNL, WordNet and WebKB are available. Alternatively, a number of tools for conceptual knowledge management have also been developed recently. These include reasoners and concept ontologies such as Mindpixel, Cyc, Learner, Freebase, YAGO, DBpedia, and MIT ConceptNet. These semantic reasoners and ontology databases can be directly exploited for applications in robotic manipulation.

The most significant of semantic knowledge acquisition systems for robotic vision systems is KnowRob (Knowledge Processing for Robots) [13], which uses reasoners and machine learning tools such as Prolog, Mallet and Weka, operating on ontology databases such as researchCyc and OMICS (indoor common-sense knowledge database). In the case of KnowRob, the data for the knowledge processing stems from three main sources: semantic environment maps, robot self-observation routines and a full-body human pose tracking system. Extensions to KnowRob, such as the K-COPMAN (Knowledge-enabled Cognitive Perception for Manipulation) system [14], enable autonomous robots to grasp and manipulate objects.

All the above frameworks for knowledge acquisition based object grasping and manipulation suffer from the fact that they require the use of explicit model databases containing object instances of the query to be processed, in order to obtain successful object recognition. K-COPMAN, for instance, uses CAD for matching 3D point clouds in order to identify the queried object in the given environment. Furthermore, while using semantic knowledge of the scene in order to improve object recognition and manipulation, these systems are largely devoid of performing implicit goal-directed cognitive tasks such as substituting a cup for a mug, bottle, jug, pitcher, pilsner, beaker, chalice, goblet or any other unlabeled object, but with a physical part affording the ability to hold liquid and a part affording grasping, given the goal of „bringing an *empty cup*' and no cups are available in the work environment.

In order to alleviate these issues, we utilize the concept of part affordances. Gibson proposed the original idea of

affordances grounded in the paradigm of direct perception. Physical affordances define the agent's interaction possibilities in terms of its physical form [16]. For example, stable and horizontal surfaces are needed to support objects, objects need to have a brim or orifice of an appropriate size, in order to be functional as a container to drink from. Additional examples of affordances studied with respect to robotic manipulation in [16] include „sittability' affordance of a chair that depends on body-scaled ratios, doorways affording going through if the agent fits through the opening, and monitors afford viewing depending on lighting conditions, surface properties, and the agent's viewpoint. The spectrum of affordances have been extended to include social-institutional affordances, defining affordances based on conventions and legally allowed possibilities leading to mental affordances. Affordances based on History, Intentional perspective, Physical environment, and Event sequences (HIPE) leading to functional knowledge from mental simulations have been studied in [15]. Affordances serve as key to building a generic, scalable and cognitive architecture for visual perception. „Affordance based object recognition' or recognition based on affordance features is an important step in this regard.

II. OVERVIEW

The primary contribution of this paper is in providing a scalable knowledge assimilation and deployment framework for robotic grasping that is free of 3D model instance representations. The second contribution of this paper is the introduction of „*Conceptual Equivalence Classes*' and their unique definition in terms of the minimalistic features of Part Functional Affordances and Part Grasp Affordances, leading to implicit cognitive processing for successful goal attainment. The third main contribution is in providing a practical pathway for symbol binding – from concepts to observables by defining functional geometry mappings. A fourth contribution is the automatic generation of grasp points, knowledge of affordance and interaction modes for unknown/ un-modeled objects based on partial information obtained from the constituent parts. Other contributions include algorithms for part detection and segmentation from range images, a scalable architecture for grasping that can be extended with textual, conceptual, visual (2D/3D) model databases.

As stated earlier, the focal point of our grasping system is based on the concept of part affordances. Recognition by components (RBC) of objects has been a significant framework in cognitive vision [6, 7, 8]. This theory expounds the use of known part shapes (called as geons) towards object recognition. In this paper, we generalize this theory towards „*Equivalence Class*' recognition using part affordances. These „*Conceptual Equivalence Classes*' help define the scalable and generic nature of the Visual Cognitive Engine. This system has been built with the target of enabling a robotic arm to grasp a-priori unseen objects in a cluttered scene using global knowledge from ontology bases and part shape detection. The Visual Cognitive Engine uses inputs from multiple ontology bases – ConceptNet (for unit concept definitions), WordNet/WebKB (for textual unit definitions), Otto Bock Human Grasping database (for grasp affordances), together with the Part Functional Affordance Schema, developed in this paper, to establish semantic unit affordances. ImageNet and other image feature resources (for visual unit definitions) can also be used in our framework, though this is not important from the standpoint of grasping. While the original approach towards RBC uses Dynamic Link Architecture (DLA) for learning geon assemblies [7] and hence objects, recent algorithms such as Attributed Graph Matching [17] and advanced variations of it such as the Elastic Graph Dynamic Link Model (EGDLM) have proven to be highly successful in recognition of objects or segments based on descriptions of these entities as a combination of nodes representing parts or patches in a graph. In our framework, we use a practical and fast variation of Attributed Graph Matching for both symbolic and metric nodes using the Hungarian algorithm from [18] to perform unit – part/ part-assembly matching. Recognition of parts or part assemblies based on affordances, using the Visual Cognitive Engine enables knowledge of affordance and interaction modes for the entire object. This leads to a goal-directed object recognition and manipulation system that can perform implicit cognitive tasks such as substitute a cup for a mug, bottle, jug, pitcher, pilsner, beaker, chalice, goblet (based on textual equivalency descriptors) or any other unlabeled object, but with a physical part affording the ability to hold liquid and a part affording grasping (using conceptual equivalency classes), given the goal of „bringing an empty cup' and no cups are available in the work environment. The performance of such a system is superior to traditional view-point based object recognition and manipulation systems.

III. CONCEPT BUILDING

A. Conceptual Equivalence Classes

The fundamental basis of our framework revolves around the theme of „Conceptual Equivalence Classes'. We define these classes as sets of objects that are interchangeable from the view-point of usage for the primary functionality of the object. Hence, objects such as mugs, cups and beakers form an equivalence class. Bags and baskets also form an equivalence class, so do cans and bottles, bikes and motorbikes and so forth. We hypothesize here that all equivalence classes can be uniquely defined and recognized in terms of their (a) Part Functional Affordance Schema and (b) Part Grasp Affordance Schema. The Part Functional Affordance Schema is explained in detail in section III.F, while Part Grasp Affordances are explained in section III.E. These schemas define the structural functionality of the parts and the structural grasp-ability respectively. Recognition of conceptual equivalence classes is analogous to generic and cognitive object recognition systems that have been studied in [6, 7, 8]. It should be noted here that the definition of conceptual equivalency class used here is distinct and unrelated to the equivalency class definitions provided by the OWL framework, which uses only textual or named entity equivalency.

B. Textual Unit Definitions

In our framework, we employ WordNet [5] for generating textual unit definitions for concepts or objects queried for. While WebKB provides improvements over WordNet, while returning results that are restricted to nouns (of specific interest to our framework), the standalone nature of WordNet recommends its usage. WordNet provides a lexical database in English with grouped sets of cognitive synonyms (synsets), each expressing a distinct concept. It also records the various semantic relations between these synonym sets, such as hypernyms (higher level classes), hyponyms (sub-classes), coordinate terms (terms with shared hypernyms), holonyms (encompassing structure) and meronyms (constituent parts). The system interacts with the WordNet interface based on the queried term to obtain a possible match. This is discussed in detail in section III.G. The system also assimilates concept 3D geometric shape information such as Sphere, Cylinder, Cube, Cone, Ellipsoid, Prism, etc., 2D geometric shape information such as Square, Triangle, Hexagon, Pentagon, Ellipse etc. and abstract structural concepts such as Thin, Thick, Flat, Sharp, Convex, Concave etc. by parsing the concept definition. Additionally, information on material properties of the concept such as Metal, Wood, Stone, Ceramic etc. and part functional affordance properties (based on terms such as Cut, Contain, Store, Hold, Support, Wrap, Roll, Move, Ride, Enter, Exit, Gap, Hole) are also obtained and stored by the system.

C. Conceptual Unit Definitions

For the case of conceptual unit definitions, we employ the Open Mind Common Sense (OMCS) [11] based ConceptNet framework. ConceptNet has been used in the context of robotic task management [12]. The particular choice of this ontology database is due to its exhaustiveness, ease of use and suitability of attributes with respect to our affordance framework. The ontology provides English language based conceptual groupings. The database links each concept with properties such as „InstanceOf' and „SymbolOf' – possibile semantic replacements, „ConceptuallyRelatedTo' – possible functional/conceptual replacements, „PartOf' – encompassing structures, „ReceivesAction', „CapableOf', „UsedFor' – possible functional affordances as well as „MadeOf', „HasProperty' etc. that provide further information about the concept. The use of these properties enables the part affordance based equivalence class selection. Detailed description on the querying process using these tags is presented in section III.G.

D. Visual Unit Definitions

While visual unit definitions can be used to improve the performance of the system or to obtain instance level recognition, our novel framework for conceptual equivalence class recognition and grasping system does not require the use of these databases and hence is 3D/2D model free. Furthermore, it should be noted that from the viewpoint of grasping using range images, monocular image information is largely superfluous. Instance level recognition, if necessary in future revisions to the system, can be carried out using a bag of features approach working with SIFT/SURF or other state-of-art feature descriptors on labeled image or 3D shape databases (such as LabelMe, LabelMe 3D and ImageNet).

E. Grasp Affordance Definitions

For the case of part grasp affordance definitions, a number of systems are available. These can be used for limiting the large number of possible hand configurations using grasp preshapes. Humans typically simplify the task of grasping by selecting one of only a few different prehensile postures based on object geometry. One of the earliest grasp taxonomy is due to Cutkosky [4]. In our system we employ the „Human Grasping Database' [3] from KTH-Otto Bock. This taxonomy lists 33 different grasp types hierarchically assimilated in 17 grasp super-types. It is possible to most of these grasp types to geometric shapes they are capable of handling. A representative set of grasp affordances from the database are presented in Fig 1. Each query concept is defined (as a whole or in parts) to provide grasp affordances of the types listed in the taxonomy database.

Nr.	Name	Picture	Type	Opp. Type	Thumb Pos.	VF1	VF2	VF3
6	Prismatic 4 Finger		Precision	Pad	Abd	1	2-5	
10	Power Disk		Power	Palm	Abd	P	2-5	
19	Distal Type		Power	Pad	Abd	1	2-5	
22	Parallel Extension		Precision	Pad	Add	1	2-5	

Figure 1. Representative Grasp Affordances from Otto Bock Human Grasping Database

F. Part Functional Affordance Schema

The most important component of the presented system is the Part Functional Affordance Schema. This component essentially performs the symbol binding – mapping concepts: in our case – the *Conceptual Equivalence Classes* to visual data in the form of 3D geometries. While various schemes for affordance definitions have been studied in the past, we utilize a set of part functional affordance schema, largely with respect to objects found in households and work environments. These affordances are based on functional form fit of the Conceptual Equivalence Classes. A representative section of the part functional affordance schema is presented in Table I. Note that the functional affordance here is defined with respect to objects of the class being able to perform the defined function.

TABLE I. REPRESENTATIVE PART FUNCTIONAL AFFORDANCE SCHEMA

Part Functional Affordance	Geometric Mapping	Examples
Contain - ability	High convexity	Empty bowl, Cup
Support - ability	Flat - Convex	Plate, Table

Intrinsic contain - ability	Cylinder/Cube /Cuboid/Prism	Canister, Box
Incision - ability	Sharp edge (flat linear surface)	Knife, Screwdrive r
Engrave - ability	Sharp Tip	Cone, Pen
2D Roll - ability	Circular/ Cylindrical	Tire, Paper Roll
3D Roll - ability	Spherical	Ball
Weed - ability[a]	Linear textural structures	Comb, Brush
Filter - ability[a]	Bi-linear textural structures	Grid, Filters
Wrap(p) -ability	w.r.t. given shape	Shoe, Glove
Connect - ability[a]	Solid with support (m)	Plug, USB Stick

a. Joint Affordances

The scale of each part is also defined with respect to a discrete terminology set based on comparative sizes – (finger (f), hand (h), bi-hand (b), arm/knee (a), torso (t), sitting posture (i), standing posture (d), non-graspable (n) etc.). The conceptual equivalence classes are defined based on joint affordances of parts of the objects, along with their topological relationships. Some of the various topological relationships (for 2-part objects) used are Table II.

TABLE II. PART JOINT TOPOLOGICAL RELATIONSHIPS

Relationship Code	Details
1v2	1 vertical 2
1h2	1 horizontal 2
1v2n	1 opposition vertical 2
1h2n	1 opposition horizontal 2
1s2	1 staggered 2
1os2	1 orthogonally staggered 2

In Table II, 1 indicates the larger object and 2 the smaller one, vertical dimension refers to the smallest of the 3 dimensions and horizontal to the largest. All relationships are with respect to the non-symmetrical axis of the object (for e.g. the opening in a roughly cuboidal bag). Opposition refers to the relationship with respect to the face opposite to the non-symmetrical face.

Based on these attribute definitions, the equivalence classes can be uniquely represented. Examples of equivalence classes are provided in Table III. Note that (ga) denotes grasp affordance and (pa) denotes part affordance.

TABLE III. EXAMPLE EQUIVALENCE CLASS DEFINTIONS

Equivalence Class – Represented by its Dominant Member	Defintion
Bag	**1v2, b-a, handle (ga), opening (pa: containability)**
Plate	**h-b, (ga), (pa: supportability)**
Cup	**1h2, f-h, handle (ga), opening (pa: containability)**
Chair	**1os2, a-i, 2x(pa: supportability)**
Canister	**h-b, (pa: intrinsic containability)**
Box	**h-i, (pa: intrinsic containability)**
Plug	**1v2n, f-h, support, contact (pa: connectability (m))**
Knife	**1h2, f-h, grip, blade (pa: incisionability)**
Bike	**b,a,a, 1v2(3hv4), seat (pa: supportability), 2xwheels (pa: 2drollability)**
Laptop	**b-a, (pa: supportability)**
Pen	**f-h, grip, tip (pa: engravability)**
Ball	**h-a, (pa: 3drollability)**
Spoon	**1h2, f-h, grip, opening (pa: containability)**
Spatula	**1h2, f-h, grip, opening (pa: supportability)**
Faucet	**1h2, f-h, pipe, orifice (pa: filterability)**
Suitcase	**1v2, b-a, handle, box (pa: intrinsic containability)**
Desk	**a-d (pa: supportability)**
Cabinet	**a-d (pa: intrinsic containability)**
Stair	**nx(pa: supportability)**
Shoe	**opening (pa: containability), (pa: wrappability/ ellipsoid)**
Key	**1v2n, f-h, support, contact (pa: connectability (m))**
Brush	**grip, bristles (pa: weedability)**
Shelf	**nx(pa: supportability)**
Scissors	**2xblade (pa: incisionability)**
Cars	**4xwheels (pa: 2drollability) (intrinsic containability)**

G. Query Evaluation

For any given query term, the system checks for availability of concept definition in the following list of attributes in a sequential order. The first database to be queried for is (a) the Part Affordance Schema. If unavailable, the system checks for the availability of a concept in the Part Affordance Schema that is matched using (b) the synsets of the queried term, followed by the „InstanceOf' and „SymbolOf' properties from ConceptNet, if necessary. If a match is not found, the system tries to use (c) the ConceptuallyRelatedTo property returned by ConceptNet (in response to the query term) to define possible alternatives for the object to be found. Alternatively, (d) the coordinate terms of queried object are searched for in order to obtain a conceptual replacement object. If a match is still not found, the system searches in (e) the holonym list and (f) the „PartOf' list from ConceptNet. This is followed by matching for (g) „ReceivesAction', „CapableOf', „UsedFor', which denote possible functional equivalency of the objects.

The frequency scores on each of these properties are also returned as a measure of confidence in the object found. If no matches are found in the Part Affordance Schema for the queried object or any of the alternatives to be searched for, as suggested by the above list of related objects, the system parses the definitions of the queried object returned by both WordNet and ConceptNet to search for structural properties associated with the object. These include shape geometry information such as cylindrical, spherical or cuboidal or its

alternate surface forms as well as abstract geometrical property terminologies such as flat, thick, thin, concave or convex.

Material properties of the object from the parsed definitions such as wood, stone or metal, (as well as those returned by the „MadeOf' property from ConceptNet) as well as functional affordances from WordNet are stored as properties of the concept being queried for. While it is possible that the given range scene can be searched for the required object entirely based on the geometry information or the defined geometries (from the Part Functional Affordance Schema) based on a matched affordance property returned from parsing the concept definitions, the confidence level (based on frequency scores and weighted by property confidence measures) returned by such an unit recognition scheme is very low. Furthermore, based on a learned appearance database of different material types (such as wood, stone or metal), the classification can be improved if monocular scene imagery is also available. Such a material classification approach can also be used to select salient regions in the scene in order to reduce computation requirements of the range image processing.

IV. Symbol Binding - Concepts To Observables – Functional Geometry Mapping

Symbol binding for grasping has been challenging problem. Approaches such as [23] and [24] use various heuristic methods to solve the problem. Another approach using shapes and contours combined with grasp point learning is found in [25]. Alternate approaches that aid in obtaining grasp points directly include [19, 20, 21, 22]. In our framework, we use a more elegant approach of structural definitions of functional affordances to address the problem.

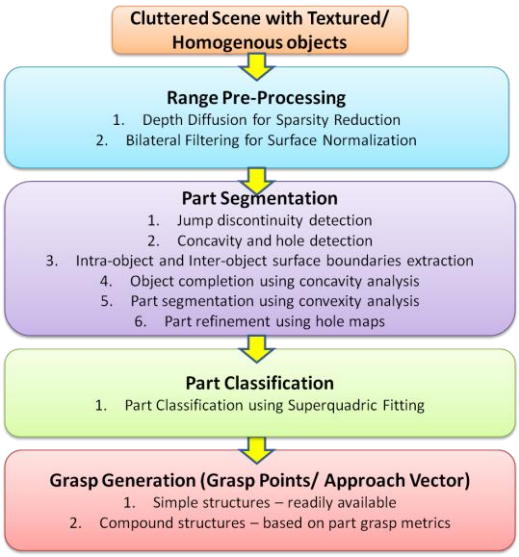


Figure 2. Pipeline for Equivalence Class Detection and Grasp Generation from Range Images

A. *Range Data Pre-Processing*

For the detection of part boundaries, it is necessary to use depth maps and surface normals obtained from the depth maps. However, the presence of noise in the system and sensor resolution can severely hamper the quality of surface normals obtained. Hence it is necessary to perform surface regularization before part detection. We employ a two step process – (a) Depth Diffusion in order to estimate depth values in regions with sparse range data (varies based on sensor used – due to reflectivity of surfaces and presence of texture), (b) Bilateral filtering for surface regularization. While the diffusion process can also regularize the surface, we keep this stage independent since both outputs are necessary for further processing and better control of scale is obtained using independent stages. Depth diffusion is required to reduce the sparsity of the point cloud for reliable and coherent surface estimation. Diffusion of depth values is carried out using a Piecewise Isotropic Laplacian Partial Differential Heat Linear Equation (PDE) Solver. By combining Multi-grid and Iterative Back Substitution (IBS) schemes to solve the PDE equation, rapid convergence is obtained.

The PDE representing the flow of heat in a 2 dimensional isotropic medium is given by

$$\frac{\partial u(\mathrm{r},t)}{\partial t} = c\left(\frac{\partial^2 u(\mathrm{r},t)}{\partial x^2} + \frac{\partial^2 u(\mathrm{r},t)}{\partial y^2}\right)$$

The equation can be reduced to a system of equations forming a block-tridiagonal matrix system with fringes. Denoting the upper tri-diagonal as $c_1(i,j)$ and lower tri-diagonal as $a_1(i,j)$ and the upper fringe as $c_2(i,j)$, lower fringe as $a_2(i,j)$, and the main diagonal as $b_1(i,j)$, The fast IBS based Depth Diffusion algorithm [9] is given by,

FringeTriDiagSolver := {InitializeSolution,
InitializeMatrixComputation, i_{iter} *-> 0,*
While[{CurrEps > EpsTol && i_{iter} *< MaxItr && AbsErr > AbsErrTol},{*
i_{iter} *->* i_{iter} *+ 1,*
StorePreviousResult,
ForwardSubstitution,BackwardSubstitution,
ComputeMaximumResidual}] }

where, *InitializeMatrixComputation* estimates the values of intermediate matrices G, Q_i, $P_i$as,

$$G(i,j) := 1/(-a_1(i,j) * Q_1(i-1,j) - a_2(i,j) * Q_2(i,-1) - b_1(i,j));$$
$$Q_1(i,j) := G(i,j) * (a_2(i,j) * Q_1(i,j-1) * Q_2(i+1,j-1) + c_1(i,j));$$
$$Q_2(i,j) := G(i,j) * c_2(i,j);$$
$$P_1(i,j) := Q_1(i,j) * X(i+1,j);$$
$$P_2(i,j) := Q_2(i,j) * X(i,j+1);$$

ForwardSubstitution and *BackwardSubstitution* modules are iterated until convergence of X estimated as,

$$M(i,j) := G(i,j) * (a_1(i,j) * (M(i-1,j) + P_2(i-1,j) + P_3(i-1,j)) + a_2(i,j) * (Q_1(i,j-1) * (M(i+1,j-1) + P_1(i+1,j-1)) + M(i,j-1) - S(i,j));$$
$$P_1(i,j) := Q_1(i,j) * X(i+1,j);$$
$$P_2(i,j) := Q_2(i,j) * X(i,j+1);$$
$$X(i,j) := M(i,j) + P_1(i,j) + P_2(i,j) + P_3(i,j)$$

where G is an inverse matrix, Q_i , P_i , M are intermediate matrices and S is the solution matrix (the right side of the equation)

Traditional isotropic diffusion solvers smooth out edge regions. We suppress the calculation of the forward and backward substitution modules for known depth pixels, thereby propagating and preserving segment boundaries as well as depth edges across iterations.

The pre-processed depth images are then surface regularized for normal estimation using an edge sensitive bilateral filter employing a Gaussian kernel. By appropriate selection of spatial and range scales, regularized depth surfaces are obtained. The equations for the bilateral filter are given by

$$\boldsymbol{h}(\boldsymbol{x}) = \frac{1}{k}\sum\sum \boldsymbol{f}(\rho).c(\rho-\boldsymbol{x}).s\left(\boldsymbol{f}(\rho)-\boldsymbol{f}(\boldsymbol{x})\right)d\rho$$

$$k(\boldsymbol{x}) = \sum\sum c(\rho-\boldsymbol{x}).s\big(\boldsymbol{f}(\rho)-\boldsymbol{f}(\boldsymbol{x})\big)d\rho$$

Where the Gaussian kernels based on the spatial difference $\rho-\boldsymbol{x}$ and the range difference $\boldsymbol{f}(\rho)-\boldsymbol{f}(\boldsymbol{x})$ establishes edge sensitive filtering.

B. Part Detection from Range Images

The majority of methods for part detection are computationally intensive. These include the relaxation labeling approach in [28]. Given the need to meet real-time constraints for deployments on robots, we use a low complexity multi-scale edge analysis scheme on the depth/disparity map. This part/object segmentation scheme links edges found at various scales using proximity and similarity measures to form enclosed regions or segments in depth maps. These edges correspond to partial object boundaries or jump discontinuities. In order to estimate the remaining object boundaries and intra-object part boundaries, we use the surface regularized depth map in order to estimate local average surface normals. A multi-scale edge analysis on the component normal images yields object contact boundaries and part boundaries. The angles between the normals at the edges are then used to classify the boundaries as contact (convexity – less than 90 degrees) or intra-object surface extremity (concavity –greater than 90 degrees) edges. Please note that in order to ensure scalability of the system to cluttered environments, we make no assumptions on the presence of a table or stable plane in the scene; detection of which is commonly done in order to greatly simply the segmentation process. Results from detection of holes and concavities in the objects are further used to enhance the object grouping. Finally, a convexity analysis is used to identify parts belonging to each object. The detected parts are then fit to geometric primitives using superquadrics. Handles and thin flat surfaces are excluded from the analysis.

C. Part Recognition using Superquadric Fitting

Superquadrics serve as highly efficient generic geometric primitives in order to obtain grasp configurations for parts/ objects with no a-priori model knowledge. Superquadrics can model superellipsoids as well as supertoroids [26, 27]. Most typical symmetrical 3D geometries – such as cubes, cones, cylinders, spheres, cuboids etc. can be modeled using superquadrics. However, super-quadrics are not very efficient in modeling concavities. Hence, we restrict the parameter values of the superquadric fitting process to only convex structures. This serves as an added advantage in geometric part affordance detection, as discussed in section IV.D. Noise and sparsity of the 3D point cloud generated can be serious issues in the fitting process. These are resolved in the range pre-processing step. The selected data points are then resampled for use with the superquadric fitting. The convergence rate of the superquadric fitting depends on the minimality of the data size. Furthermore, it is necessary to have a uniform sampling rate in the 3D space of the object. However, the number of data points on surfaces that are tangential to the camera viewpoint is typically very low. In order to alleviate these issues, a content adaptive point-cloud importance resampling based on curvature in depth has been used. Superquadrics can be represented by the following implicit equation:

$$\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}}+\left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}}+\left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}}=1$$

where ϵ_1 and ϵ_2 are squareness parameters and define the transition from a smooth curvature (as in the case of a sphere) to sharp edges (as in the case of a cuboid); a_1, a_2, a_3define the scale of the superquadric along the *x, y, z* dimensions. The fitting of the superquadric is based on the error metric – the inside-outside function (*F*) that evaluates whether a point is inside or outside or on the surface of the superquadric. The error metric is conventionally made independent of ϵ_1, the shape of the superquadric in order to obtain rapid convergence.

$$F^{\epsilon_1}(x,y,z)=\left(\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}}+\left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}}+\left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}}\right)^{\epsilon_1}$$

Furthermore, in order to normalize the convergence rates and directions, the scale factors $a_1a_2a_3$ are introduced in the error metric, resulting in the fitting function,

$$F_s(x,y,z)=\sqrt{a_1a_2a_3}(F^{\epsilon_1}(x,y,z)-1)$$

The final error metric to be minimized for the superquadric fitting is given by

$$\min_{\Lambda}\sum_{i=1}^{n}\left(\sqrt{\lambda_1\lambda_2\lambda_3}(F^{\epsilon_1}(x_i,y_i,z_i;\lambda_1,\lambda_2,\ldots,\lambda_{11})-1)\right)^2$$

where, λ_i are the parameters of the superquadric. Superquadric based 3D point cloud data approximation can be extremely efficient in the identification of stable grasp points. This enables a continuous space parameterization of objects in the scene. These parameters will form the feature vectors for the classification of geometric primitives that serve as discrete space parameterizations. The superquadrics fitting process is accomplished using a Particle Swarm Optimization (PSO) operating on a constrained superquadric equation parameterized as size variables, squareness parameters, coordinate transformation and rotation, tapering and bending parameters - a total of 15 parameters. For most practical scenes,

it was sufficient to carry out the fitting process using only 11 parameters, by excluding the bending and tapering forms. Fitting of super-quadrics (based on 15/11 parameters) to pruned 3D data is a relatively easier task due to quantitative nature of the representation. The suitability of initial conditions is very important for rapid convergence. In order to identify the geometric primitive from the superquadric parameters, classification descriptions of 8 basic primitive types, namely Sphere, Cube, Cuboid, Regular Cylinder, Cone, Prism, Pyramid, and Frustum were established. By using a combination of bounds and ranges for each of the super-quadric parameters the geometric primitives are uniquely identified.

D. Geometric Part Affordance Detection

As discussion earlier, the Part Functional Affordance Schema defines unique symbol binding from affordance concepts to observables in terms of functional geometry mapping. While certain affordances are defined based on geometrical shape structures such as cylinders, cubes, cuboids and spheres or continuous space parametric variations of these shapes (as defined by superquadrics), other affordances are defined in terms of abstract geometrical attributes such as flat, concave, convex, sharp tip, sharp edge, linear textural structures, bi-linear textural structures. Joint affordances are defined in terms of more than one part. While detection results of the first set (geometrical shape structures) is directly available from the superquadrics, results for the second set (abstract geometries) can be inferred from the superquadrics. Since superquadrics model objects or parts as convex structures, presence of a concavity (such as the open cylindrical portion of a cup) can also be verified using visibility tests for cloud points and normals (for e.g. belonging to the inner surface of the cup, in comparison with a solid cylinder). Other attributes such as flatness and sharpness, linear and bi-linear textures can also be roughly estimated based on measures of size, shape and thickness of the quadric.

E. Geometric Grasp Affordance Detection

Most of the grasp affordances based on the Otto Bock Grasping Database, can be uniquely represented in terms of geometrical shapes. For e.g., the small diameter affordance can be structurally defined as a superquadric with a high linear dimension value along one axis and small diameters along the others. This also holds true of prismatic affordance, though the diameter is much smaller. Power disk is suited for disk type structures of the size of the palm, parallel extension for cuboidal structures and distal for objects with disjoint ring shaped parts.

F. Conceptual Equivalence Class Object Selection using Attributed Graph Matching

In the given scene of interest, the queried object for the given task is found using attributed graph matching of the concept node built for the query with all geometrical objects found in the scene. Among the several attributed graph matching approaches [17, 18] available, we use a low complexity approach based on Heterogeneous Euclidean Overlap Metric (HEOM) using the Hungarian Algorithm [18] for the matching process. Each object in the scene is represented as a graph with its parts defining nodes along with vector attributes that may be symbolic (such as affordances) or metric (scales). Given the limited number of objects in a given scene, the matching process is fast and accurate. In the case that more than one object is found in the scene, the nearest object is selected for manipulation.

G. Grasp Points Generation

The final step in the pipeline is the generation of grasp points. For a given embodiment, the best set of grasp points for simple geometric primitives is well established. For the case of superquadric structures that do not fit into one of the shape descriptions, we use the closest match. For a two finger Otto Bock hand, the following grasping schema is defined:

Cubes/ Cuboids: Cylinder pregrasp shape such that the two fingers contact opposite faces. The palm should be parallel to the face orthogonal to the two opposing faces.

Spheres: Spherical pregrasp shape with the palm approach vector passing through the center of the sphere.

Cylinders/Cones: Based on the initial pose and size of the cylinder, it can be grasped from the side, or from either end.

(a) Side Grasp: Cylindrical pregrasp with the approach vector perpendicular to the side surface.

(b) End Grasp: Spherical pregrasp shape with approach vector perpendicular to end face.

For the case of cones, depending upon the size of the cone, an end grasp may be more stable.

Additional parameters such as number of parallel planes, divisions of 360 degree, grasp rotations and 180 degree rotations [1] together with constraints on time and grasp accuracy or learning of grasping modes from knowledge bases [2] can be used to decide the grasping points.

V. RESULTS AND EVALUATION

The performance of the concept evaluation and range image processing algorithms for a given scene is demonstrated using a set of queries.

The results for performance of individual range processing modules are shown in Fig. 3 and Fig. 4. The first scene (Fig. 3) is composed of 2 large mugs and 2 ping-pong paddles and the second scene (Fig. 4) is composed of 2 bags. The results of range image pre-processing are presented in Fig. 3C/D and Fig. 4C/D. Depth diffusion results are shown in Fig. 3C and 4C, while the depth normals after surface regularization are shown in Fig. 3D and 4D. In Fig. 3 and Fig. 4, E depicts results of concavity detection. Hole detection results are shown in Fig. 4F (there are no stable holes in Fig. 3). Intermediate object candidates are shown in Fig. 3F and 4G. Final object detection results are shown in Fig. 3G and 4H, followed by part detection results in Fig. 3H and 4I. The regularized scene is shown in Fig. 3I and 4J. In these images, thin strips identified as parts of objects (with holes separating them from the main object) are classified as handles, based on shape and the grasp affordances and the grasp points/approach vector for entire object are defined as creating a closure across the handle through the encompassing cavity. Besides handles, thin flat structures in the scene (for example, the surface of the ping pong paddles) are also excluded from the geometric

primitive estimation process. Fig. 3J shows the results of superquadric fitting corresponding to the cylindrical structures of mugs in the scene. Side grasp (lilac) and end grasp (green) points are generated and depicted in the figure for a sample mug. In the second scene, Fig. 4K depicts the results of superquadric fitting (cuboid). While the scale of the objects in the scene precludes direct grasping, the presence of handles provides a mechanism for grasping of these objects. Selection of the correct object for grasping (from among bags, mugs, ping-pong paddles etc.) is determined by the query input to the system.

For the first scene (Fig. 3), a search query for „jug' is presented. It should be noted that the query „jug' is not available in our equivalence class database, hence causing the search to be non-trivial. Using WordNet based parsing, renders the part affordance of „containability' with a weight measure of 2 (out of 10), based on frequency scores for primary (from definition text) and secondary characteristics (from other attributes). ConceptNet also renders the „containability' affordance along with a „HasA' attribute of „handle' which provides the grasp affordance for the given case. The attributed graph for the given query is simple and is composed of nodes for „containability' part affordance and a „handle' – small diameter grasp affordance with an overall weighted confidence score of 1.66/4 (using concept and textual unit definitions of 1 and 3 respectively). The range image processing algorithms yield both the mugs in scene as results (prioritized by the closest object), since these objects contain concavities (affordance: containability) and handles (grasp affordance) that match the query graph attributes exactly (normalized HEOM score of 1).

For the second scene (Fig. 4), a search query - „bag' is presented. Again, since no equivalence class has been defined for the term „bag', the computation of the search is non-trivial. For the given case, WordNet and ConceptNet render the „containability' affordances along with the „handle' grasp affordance. In addition, ConceptNet renders the scale parameter to be „large' and equivalent to that of a „box'. The confidence score on the resulting affordance description is 3.64/4 (since WordNet returns a high frequency score of 8). Since the queried scene contains 2 true „bags', the range processing algorithms return both the bags as query results. Again the normalized HEOM score is 1, indicating a perfect match for known attributes. It can also be seen that the confidence in the result is high for the second scene, as compared to the first, since the rate of occurrence of the object in typical scenes (reflected in the frequency score from WordNet) is higher.

VI. Conclusion and Future Work

In this paper, we have presented a scalable knowledge assimilation and deployment framework for robotic grasping that is free of 3D model instance representations. We have also introduced the paradigm of „*Conceptual Equivalence Classes*' and uniquely defined them in terms of the minimalistic features of Part Functional Affordances and Part Grasp Affordances, leading to implicit cognitive processing for successful goal attainment. We have also provided a practical pathway for symbol binding – from concepts to observables by defining functional geometry mappings. The system is also capable of automatic generation of grasp points, knowledge of affordance and interaction modes for unknown/ un-modeled objects based on partial information obtained from the constituent parts.

Currently, the number of part functional affordances supported by the system is quite limited. We plan to extend the number and range of the supported functional affordances in the future. This would also necessitate more advanced algorithms for the attributed graph matching. Furthermore, the current system is geared towards robotic grasping and manipulation while being capable of functional class level object recognition. As such, it uses only range information for the processing, without the need for 2D/3D databases. Extension of the scheme to perform instance level object recognition will necessitate the use of these databases. Moreover, while current system has been evaluated on a stand-alone system, actual deployment of the system on a robot with an arm and gripper for grasping is ongoing research. Finally, while the current system is intended to serve as a core component for goal-directed object recognition and manipulation, it can be used in a more holistic system for semantic visual perception such as the K-COPMAN.

Acknowledgment

This work is partly funded by the EU IST-FP7-IP-215821 GRASP project.

References

[1] Automatic grasp planning using shape primitives, AT Miller, S Knoop, HI Christensen, PK Allen - IEEE International Conference on Robotics and Automation, 2003

[2] Efficient and effective grasping of novel objects through learning and adapting a knowledge base, Curtis, N. Jing Xiao , Intelligent Robots and Systems, 2008

[3] T. Feix, R. Pawlik, H. Schmiedmayer, J. Romero, and D. Kragic, "A comprehensive grasp taxonomy," in Robotics, Science and Systems Conference: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation, Poster Presentation, June 2009.

[4] M. R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. Robotics and Automation, IEEE Transactions on, 5(3):269–279, 1989.

[5] Christiane Fellbaum (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

[6] Biederman, I. (1987). Recognition-by-Components: A Theory of Human Image Understanding. Psychological Review, 94, 115-147

[7] Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. Psychological Review, 99, 480-517

[8] Chaitanya Raju, Karthik Mahesh Varadarajan, Niyant Krishnamurthi, Shuli Xu, Irving Biederman and Troy Kelley, "Cognitive object recognition system (CORS)", Proc. SPIE 7692, 76920L (2010).

[9] KM.Varadarajan, M. Vincze, „Real-Time Depth Diffusion for 3D Surface Reconstruction', IEEE Int. Conference on Image Processing (ICIP), 2010.

[10] Matthias J. Schlemmer and Markus Vincze, Theoretic Foundations of Situating Cognitive Vision in Robots and Cognitive Systems. Submitted to Journal Image and Vision Computing.

[11] Catherine Havasi, Robert Speer, Jason Alonso, Conceptnet 3: A Flexible, Multilingual Semantic Network For Common Sense

Knowledge, In Recent Advances in Natural Language Processing (27--29 September 2007)

[12] EpistemeBase: a Semantic Memory System for Task Planning under Uncertainties, Xiaofeng Xiong, Ying Hu and Jianwei Zhang, IROS 2010

[13] Moritz Tenorth , Michael Beetz , KnowRob --- Knowledge Processing for Autonomous Personal Robots, 2009, IEEE/RSJ International Conference on Intelligent RObots and Systems

[14] Combining Perception and Knowledge Processing for Everyday Manipulation, Dejan Pangercic and Moritz Tenorth and Dominik Jain and Michael Beetz, IROS 2010

[15] L. Barsalou, S. Sloman, and S. Chaigneau (2005) The HIPE Theory of Function. in: L. Carlson and E. van der Zee (Eds.), Representing functional features for language and space: Insights from perception, categorization and development. pp. 131-147, Oxford University Press, New York.

[16] Martin Raubal and Reinhard Moratz, A Functional Model for Affordance-Based Agents, Towards Affordance-Based Robot Control, Lecture Notes in Computer Science, 2008, Volume 4760/2008, 91-105

[17] Irving Hofman and Ray Jarvis, Object Recognition Via Attributed Graph Matching, ACRA 2000.

[18] Salim Jouili and Salvatore Tabbone, Attributed Graph Matching using Local Descriptions, Advanced Concepts for Intelligent Vision Systems - Acivs 2009.

[19] M. Berger, G. Bachler, and S. Scherer, "Vision Guided bin Picking and Mounting in a Flexible Assembly Cell", Intelligent Problem Solving: Methodologies and Approaches, pp. 255-321, 2000.

[20] C. Dunes, E. Marchand, C. Collowet, and C. Leroux, "Active Rough Shape Estimation of Unknown Objects", In International Conference on Robotics and Automation, 2008.

[21] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics", In IEEE Transactions on Robotics and Automation., 8(3):313–326, June 1992.

[22] K.E. Ozden, K. Schindler, and L.J. van Gool, "Simultaneous Segmentation and 3D Reconstruction of Monocular Image Sequences", In International Conference on Computer Vision, pp: 1-8, 2007.

[23] Learning grasp strategies with partial shape information, Ashutosh Saxena, Lawson Wong, Andrew Y. Ng. AAAI, 2008

[24] Learning to Grasp Novel Objects using Vision, Ashutosh Saxena, Justin Driemeyer, Justin Kearns, Chioma Osondu, Andrew Y. Ng, ISER, 2006

[25] Grasping Familiar Objects Using Shape Context, J Bohg, D Kragic - International Conference on Advanced Robotics, 2009

[26] Superquadric Segmentation in Range Images via Fusion of Region and Boundary Information, Dimitrios Katsoulas, Christian Cea Bastidas, and Dimitrios Kosmopoulos, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 30, NO. 5, MAY 2008

[27] Reconstruction of Superquadric 3D Models by Parallel Particle Swarm Optimization Algorithm with Island Model, Fang Huang and Xiao-Ping Fan, ICIC 2005, Part I, LNCS 3644, pp. 757-766, Springer Verlag, 2005.

[28] Lejeune, A., & Ferrie, F.P. (1993) 'Finding the parts of objects in range images', McGill University.

[29] SRVC, http://www.semantic-robot-vision-challenge.org/

[30] RoboEarth, http://www.roboearth.org/

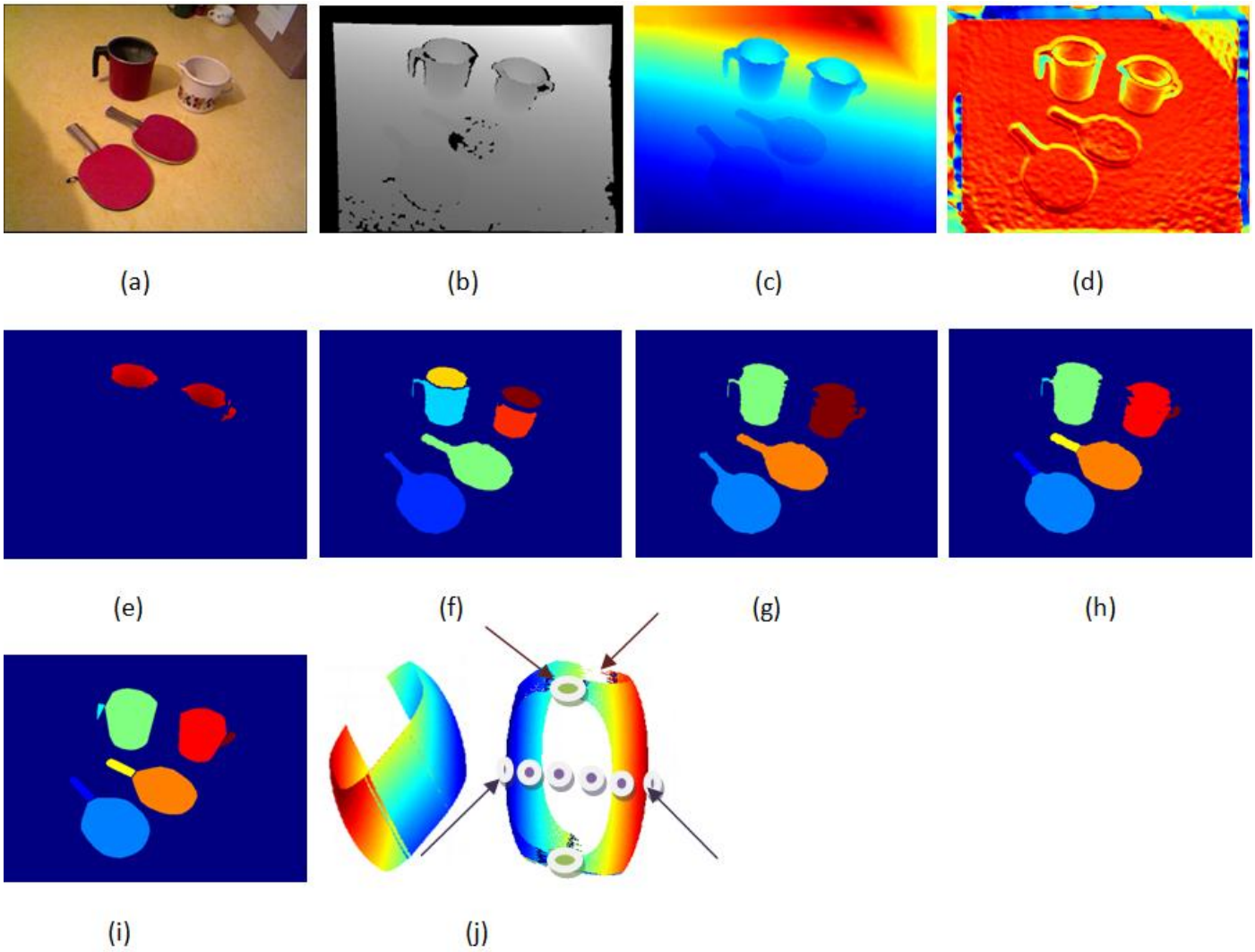


Figure 3. (a) Sample scene 1 (b) Input depth map (c) Diffused depth map (d) Depth normals after surface regularization (e) Concavity map (f) Object candidates (g) Object map (h) Part map (i) Regularized scene (j) Superquadric fitting of parts along with grasp points

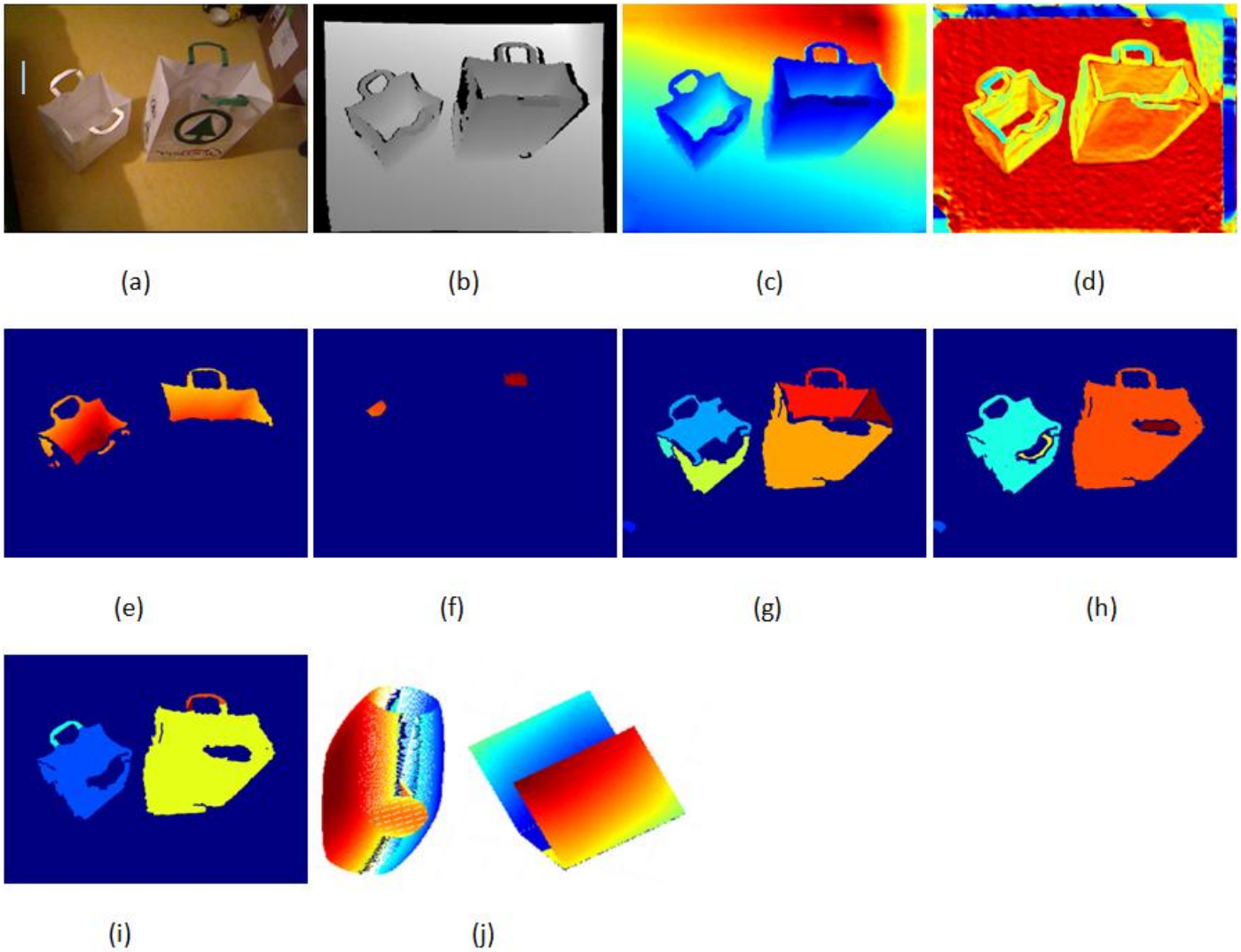


Figure 4. (a) Sample scene 2 (b) Input depth map (c) Diffused depth map (d) Depth normals after surface regularization (e) Concavity map (f) Hole map (g) Object candidates (h) Object map (i) Part map (j) Superquadric fitting of parts

Combining 2D and 3D Object Categorization with Task-Constrained Grasping

Marianna Madry, Dan Song and Danica Kragic

***Abstract*— The main objective of this work is to enable transfer of grasp knowledge between object categories, defined using geometric properties and functionality. To this end, we present an object categorization system integrated with a grasp planning system. The categorization system employs both 2D (RGB image) and 3D (extracted point cloud) information about an object. We present and evaluate several 2D-3D integration strategies. The system is tested on real data collected using an active stereo robot head, capable of vergence and foveation. The data is generated in natural scenes, for a number of household object categories. The system is built upon an active scene segmentation module, able of generating object hypotheses and segmenting them from the background in real-time. The output of categorization is used in a probabilistic grasp planning system that encodes task, object and action properties. The experimental evaluation compares individual 2D and 3D categorization approaches with the integrated system as well as and it demonstrates the usefulness of the categorization in goal-directed grasping.**

I. INTRODUCTION

Robotics poses several challenges to visual processing that go beyond the current work in the area of computer vision. The mainstream approaches in computer vision attempt to model the world through inference on 2D data, in robotic applications the aim is to allow a robot to model and understand the world by acting in and interacting with the environment. Our aim is to leverage on some recent advances on object categorization considering both 2D and 3D data and show how it can facilitate robot grasping. Fig. 1 shows our Object Categorization System (OCS), that consists of:

- a front-end with an active robot head equipped with foveal and peripheral cameras that provide input to the real-time scene segmentation system [1];
- a back-end, the probabilistic grasp reasoning system, that encodes task-related grasping [2][3].

The authors are with KTH - Royal Institute of Technology, Sweden, as members of the Computer Vision & Active Perception Lab., Centre for Autonomous Systems, e-mail: madry,dsong,danik@csc.kth.se. This work was supported by EU project GRASP, IST-FP7-IP-215821 and Swedish Foundation for Strategic Research.

We are motivated by the fact that humans classify an object according to its functionality, which is naturally linked to what kind of task it affords [4]. The grasp reasoning system uses a Bayesian network (BN) representation, trained on a number of example grasping tasks. The system models the conditional dependencies between the grasping tasks and the object class variables. We demonstrate that, by using the integrated system, the robot can not only choose the objects in a 3D scene that afford the assigned task, but also plan the grasp such that it satisfies the constraints posed by the task. Thus, grasp knowledge can be transferred between objects that belong to the same category, though the details of the geometry and physical properties may vary.

From the stereo system we have access to both 2D images and a 3D point cloud representation of the scene. Ideally, an object categorization system should be able to exploit both to improve robustness. Since each channel (2D and 3D) has different characteristics, fusion of 2D-3D categorization can provide a more comprehensive description of an object. Finally, integration may result in a system that is resilient to the degradation of one of the cues, e.g. for low textured objects in case of 3D or for lightning changes in case of 2D. In this work, we also analyze robustness of different 2D and 3D feature representations in realistic scenes, and assess their diversity to better understand the requirements for their integration.

As shown in Fig. 1 (middle column), we train a separate OCS for each descriptor/representation and then fuse evidences from a few OCSs to obtain the final categorization. Results from the extensive experiments on real data for eleven object categories show that: (a) the proposed 2D-3D object categorization system achieves high categorization rate (92%), significantly better than any of the classic single cue OCSs in the same task; (b) simple linear cue integration methods are more efficient than complex methods in case of the limited amount of data, as in our case.

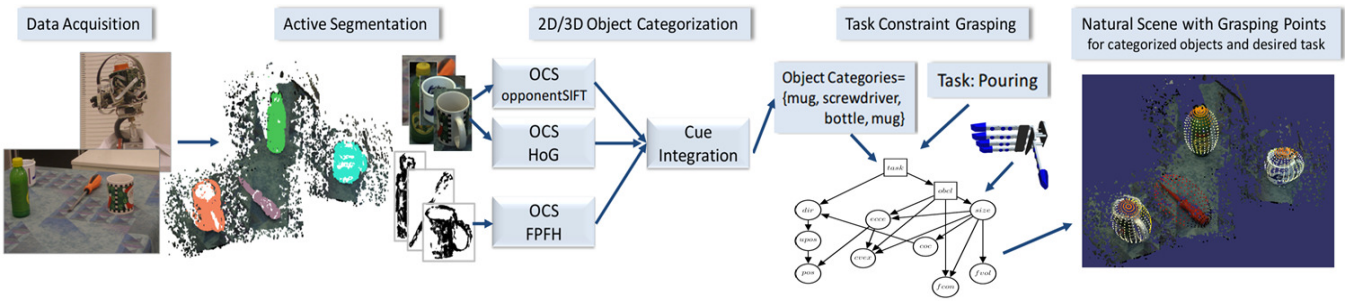


Fig. 1. From categorization to task based grasping. After data acquisition using a robot head, objects are segmented and categorized using our 2D-3D Object Categorization Systems (OCSs). Then, grasping hypotheses are generated, taking the task into account. The image is best viewed in color.

Preprint submitted to 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 28, 2011.

II. RELATED WORK

The knowledge of object categories is useful for task-related grasping: for humans, it is natural to use and manipulate objects based on their functionality or current task [4] – when pouring from an object, the fingers should not be occluding the opening of the object. For the object category that can be used to pour liquid from, the knowledge of how to grasp it to allow pouring may be transferred between objects that belong to the same category. Although partial object knowledge has been used recently in grasping applications [5][6], integration of task-related grasp planning and categorization in real scenes is a clear novelty.

In [2] and [3], we have developed a probabilistic framework using Bayesian network (BN) to represent the task-related grasping. The semantic task requirements are encoded through the conditional dependencies between a *task* variable and a set of object and grasp features. Using this *task constraint* BN, we have demonstrated that the robot is able to: (i) reason at the symbolic level, and (ii) make detailed decisions on sensorimotor level, e.g. plan grasps that afford *pouring*.

This work, however, was mostly done in a simulation environment and the inference engine assumed the object class unknown. Learning of the network structure in [3] revealed the importance of the categorical information. Thus, we integrated OCS based on real sensory data with the task constraint model using BNs. We show that in the integrated system, the robot can not only choose the objects in a scene that affords the assigned task, but also plan the grasp to satisfy task constraints.

The use of the system in real scenes demands robustness to noise and occlusions, as well as to viewpoint, illumination and resolution changes. Object categorization imposes important requirements: the object representation has to ensure accurate discrimination between object categories and, at the same time, handle high within-class variations. An important assumption, determining the usefulness of the cue integration, is that the information provided by the different cues is complementary. Intuitively, we can expect that the integration of descriptors capturing different object properties (such as appearance, color, shape) is most effective.

Several descriptors have been proposed in the field of computer vision to encode object appearance (SIFT [7], textones [8]), color (opponentSIFT [9]) and contour shape (HoG [10]). The studies on 2D cue integration [11] show that contour- and shape-based methods are adequate for handling the generalization requirements needed for object categorization but are not robust to occlusions. On the other hand, appearance- and color-based descriptors have been successfully applied in object (instance) recognition and detection [7], [8]. However, their performance drops significantly in case of clutter and illumination changes. In the field of object retrieval and computer graphics, a number of 3D shape descriptors have been proposed [12]. Only a few of them are applicable to real 3D data that covers only a visible part of the object: spin images [13], RSD [14], FPFH [15],[16]. We show that the 3D descriptors cope better with the viewpoint changes than 2D descriptors, but they are poor at discriminating categories of similar shape (e.g. *citrus*/*ball*). Integration of different descriptors significantly increases performance and robustness of our system.

Another important approach in computer vision are methods for part-based representations of objects. These methods differ with respect to the amount of spatial information they encode. Object parts can, for example, be treated as geometrically independent (*bag-of-words* BoW model [17]). Another approach may be to store only a coarse global spatial information (*spatial pyramids* [18]) or more explicit spatial information (*constellation models* [19], [20], including methods based on probabilistic modeling [21]). Our system uses the BoW and spatial pyramid methods due to their short training-time (small number of parameters to estimate) and short run-time due to the low computational complexity.

Regarding cue integration, several approaches have been applied to object recognition and categorization based on 2D data. These methods can be divided into: *low level integration* and *high level integration*. The low level integration operates on feature vectors, and the examples are mostly limited to the early work in object recognition [22] due to the curse of dimensionality [23, p.170]. The high level integration is commonly accomplished by an ensemble of classifiers or experts. The most common techniques include [24]: majority voting of classifiers [11] and methods based on algebraic combination of classifier outputs. The classifier outputs can be combined using linear [25] or nonlinear [26] combination of evidences. Our results clearly demonstrate that in case of real applications where the amount of training data is limited, the simpler linear algebraic methods are more efficient.

In the literature, there are only very few examples of combining 2D and 3D descriptors for object categorization. In the recent work [27], authors built a hierarchical OCS system in which 3D descriptor (Global RSD) is used to narrow choice of categories to those of similar shape, and then 2D descriptor (SURF) is applied. However, the generalization ability of this system is low. Its performance drastically drops from 98% when trained and tested on the same data, to 50% when tested on the new data. We present a systematic study on combining different 2D and 3D features for object categorization with application in natural scenes, revealing the challenges of real settings.

III. 2D-3D OBJECT CATEGORIZATION SYSTEM

As shown in Fig. 1, we first build a single cue OCS for each feature descriptor which are then integrated to provide the final categorization. All single cue OCSs implement the following methodology: (a) data acquisition (Section III-A), (b) feature extraction (Section III-B), and (c) classification (Section III-C). The methods used to integrate these single cue OCSs will be described in Section III-D.

A. Data Acquisition

Prior to categorization, a scene including multiple objects is first segmented using our multi-cue scene segmentation system reported in [1] (see Fig. 1). In short, the method relies

Preprint submitted to 2011 IEEE/RSJ International Conference
on Intelligent Robots and Systems. Received March 28, 2011.

on attentional mechanisms in the peripheral view to direct cameras towards regions of interest, subsequently grouping areas close to the center of fixation as the foreground. The disparity maps, computed using the Stable Matching [28], are converted into 3D points. An assumption of a supporting plane is used to provide a better segmention from the background. The points are labeled as belonging either to the object (foreground), supporting plane (flat surface) or the background. The important fact to stress here is that the system generates object hypotheses *without* relying on the learnt categories which is a common approach in the literature. Since our long term interest is also to incrementally learn new object categories, our approach is rather natural.

The segmented point cloud is further processed to remove outliers and equalize point density. We rely on the statistical outlier removal and voxel grid filters from the ROS PCL [29]. The resulting point cloud contains ca. 2000 points and represents a visible part of an object. In order to save computational time we do not reconstruct the whole object from its partial view as in Marton et al. [14][27]. Such reconstruction methods often assume objects to be symmetrical which is not always the case. The segmented parts of RGB images do not require any further pre-processing before feature extraction.

B. Feature Extraction

The choice of a proper object representation is crucial for achieving good categorization rates. Ideally, the object representation should have high discriminative power, be robust to noise, occlusions and viewpoint changes, illumination and resolution aspects. For cue integration, information provided by the different cues has to be complementary. Therefore, from a segmented part of an image, we extract multiple 2D descriptors encoding different object attributes: appearance (SIFT [7]), color (opponentSIFT [9]), contour shape (HoG [10])[2]. The final object representation for 2D descriptors follows a concept of the spatial pyramid [18]. The 3D shape properties of an object are obtained by applying the FPFH descriptor [15] to each 3D point in the segmented point cloud. It was shown that the normal-based descriptors obtain high performance for the task [16]. To obtain the final object representation, the BoW model [17] is employed.

C. Classification

For classification, we use SVMs with a χ^2 kernel, successfully applied in previous studies [9][10][27]. For the purpose of cue integration, we need information about the confidence with which an object is assigned to a particular class. Several studies were devoted to find confidence estimates for large margin classifiers [30], [25]. In principle, they interpret the value of the discriminative function as a distance of a sample to the optimal hyperplane. The closer the sample is to the hyperplane the lower is the probability (confidence) of a correct classification. In this work, we use the One-against-All strategy for M-class SVMs and the confidence measure for a sample $\mathbf{x}$ is calculated as [31]:

$$C(\mathbf{x}) = D_j*(\mathbf{x}) - \max_{j=1\ldots M, j\neq j*}\{D_j(\mathbf{x})\} \qquad (1)$$

where $D_j(\mathbf{x})$ is equal to the difference between the average distance of the training samples to the hyperplane and the distance from $\mathbf{x}$ to the hyperplane. In the preliminary experiments this approach was superior to the Platt's method [30].

D. Cue Integration

The 2D-3D object categorization system is created by integrating evidences from the single cue OCSs at the high level, i.e. after the single-cue classification is performed. We use methods based on an algebraic combination of classifier outputs since they are the most robust to noisy cues (see Section II). We evaluate both the linear and nonlinear algebraic techniques.

In case of the linear techniques, the total support for each class is obtained as a linear weighted sum, product or max function $F(\cdot)$ of the evidences provided by individual classifiers. The final decision is made by choosing the class with the strongest support. Let us assume that d_{ij} is an evidence provided by classifier i for a category j, and w_i is a weight for classifier i (both are normalized to sum up to one for all L classifiers and M categories), then the class with the strongest support $j_0 \in \{1,\ldots,M\}$ is chosen as:

$$j_0 = \arg\max_{j=1,\ldots,M} \frac{F(d_{1j},\ldots,d_{Lj};w_1,\ldots,w_L)}{\sum_{j=1}^{M} F(d_{1j},\ldots,d_{Lj};w_1,\ldots,w_L)}. \qquad (2)$$

The weights $w_i|_{i=1,\ldots,L}$ are estimated during training. In this setup, the sum rule is equvalent to the Discriminative Accumulation Scheme (DAS) proposed in [25].

In case of the nonlinear techniques, we have used an approach where an additional SVM classifier is trained to model the relation between evidences provided by the different single cue OCSs [26]. The outputs from the single cue OCSs are concatenated to build a feature vector that is fed to the subsequent SVM classifier. During the training, parameters of the nonlinear function $F(\cdot)$, that is equal to the classifier kernal function, are estimated. We have evaluated the performance of the following three nonlinear function: (a) radial basis function (RBF), (b) χ^2 function, and (c) histogram intersection.

The linear methods are simple and have low computational complexity. However, to infer weights $w_i|_{i=1,\ldots,L}$, an exhaustive search over parameter values is needed which becomes an intractable task for a large number of cues. The nonlinear methods owing to more complex function may better adapt to the varying properties of the cues. However, they also require a larger training dataset what may be unfeasible in real applications.

IV. Modeling Task Constraints

In [2], [3], we developed a unified framework for embodiement-specific grasp representation. It consists of a probabilistic graphical model, the Bayesian network (BN), and a new multi-variate discretization method. BN models the conceptual task requirements through conditional dependencies among a set of task, object, action and constraint variables. The discretization model provides a compact data representation that allows efficient learning of the conditional structures in the BN.

Preprint submitted to 2011 IEEE/RSJ International Conference
on Intelligent Robots and Systems. Received March 28, 2011.

Fig. 2. Examples of objects used to create the database presented in Section V-A. The data for all the 110 objects can be viewed at our web site `http://www.csc.kth.se/~madry/research/stereo_database/index.php`.

(a) (b)

Fig. 3. Examples of imperfect segmentation in both 2D and 3D: (a) only a part of an object is detected, or (b) the segmentation mask contains background points (background points are marked in red).

The model is trained using a synthetic database of objects, grasps generated on them, and the task labels provided by the human. The data generation is based on the toolbox BADGr [32] in the simulation environment provided by GraspIt! [33]. BADGr provides the ways of object 3D shape approximation, grasp planning, execution and also grasp-related feature extraction and task labeling. We refer the reader to the detailed process of data generation in [2], [3].

Both the structure and the parameters of the BN were trained using the task related grasp database. The BN structure encodes the dependencies among the set of task-related variables, and the parameters encode their conditional probability distributions. Figure 4 shows the learned structure of the BN, and the features represented in this BN are listed in Table I. Once trained, the model can be used to infer local distribution on each individual or small set of variables, based on partial or complete observation of others. For example, we could obtain $P(pos|task, obcl)$, the probability of the grasp position pos, given the observed object class $obcl$, and an assigned grasping task $task$.

V. EXPERIMENTAL EVALUATION

In this section, we present an experimental evaluation of our 2D-3D object categorization system integrated with the task-constrained grasp planner on the real stereo data. The description of the dataset and experimental setup is given in Section V-A and V-B. First, we study robustness of different 2D and 3D descriptors under varying viewpoint condition and their applicability to cue integration (Section V-C). Then, we present the systematic evaluation of several 2D-3D integration strategies (Section V-D). Finally, we demonstrate the results of an integrated system considering categorization for task-constrained object grasping (Section V-E).

TABLE I
FEATURES USED FOR THE TASK CONSTRAINT BAYESIAN NETWORK.

Name	Dimension	States	Description
$task$	-	4	Task Identifier
$obcl$	-	6	Object Class
$size$	3	8	Object Dimensions
$cvex$	1	4	Convexity Value $[0, 1]$
$ecce$	1	4	Eccentricity $[0, 1]$
dir	4	15	Approach Direction (Quaternion)
pos	3	12	Grasp Position
$upos$	3	8	Unified Spherical Grasp Position
$fcon$	11	6	Final Hand Configuration
coc	3	4	Center of Contacts
$fvol$	1	4	Free Volume

A. Database

Most of the databases used for categorization purposes are storing only 2D image information [34]. Although there are databases for 3D object retrieval [35], these do not contain both 2D images and 3D object structure. The KIT ObjectModels Web Database [36], contains both 2D and 3D data, but was created for a purpose of object (instance) recognition and contains only a small number of simple shape categories [27]. Moreover, for each shape category a number of models is small and a natural variability of object appearance and shape within each category is not well represented.

For these reasons, we created a new database that contains 11 object categories: *ball, bottle, box, can, toy-car, citrus, mug, 4-legged toy-animal, screwdriver, tissue and tube*, each with 10 different object instances per category (in total 110 objects, examples of objects for each category are presented in Figure 2). Different objects were chosen for each category in order to capture variations in appearance, shape and size within each class. For each object, the 2D (RGB image) and 3D (point cloud) data were collected in 16 different views around an object (separated by 22.5°) using the 7-joint Armar III robotic head with foveal and peripheral stereo cameras, see Figure 1. To differentiate an object from a background we used the active segmentation method [1] that generated good results in ca. 90% of cases. For some object categories, such as *toy-car*, *toy-animal* and *screwdriver*, segmentation was more challenging, see Figure 3.

B. Experimental Setup

The database was divided into four sets used for: (1) training, (2) validation of OCS parameters, (3) validation of the cue integration parameters, and (4) testing. Objects were randomly selected for each set with the ratio 4:1:1:4 objects per category. In total, data for 44 objects were used for

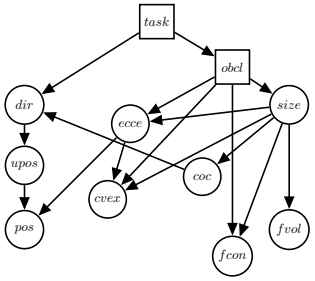


Fig. 4. The structure of the Bayesian network task constraint model.

Fig. 5. *Setup-50*. Objects from eight different viewpoints selected to train the system (top row) and evaluate its performance (bottom row). The test set contains 50% data collected from the different viewpoints than the training data.

TABLE II

SETUP FOR EXPERIMENTS IN SECTIONS V-C AND V-D.

Descriptor	Parameters
SIFT	grid detector, spacing=6px, L2-normalized
opponentSIFT	grid detector, spacing=6px, L2-normalized
HoG	Canny detec., angle range$\in (0, 180)$, #bins=20
FPFH	no_subdiv=11, kSearch=50[1]

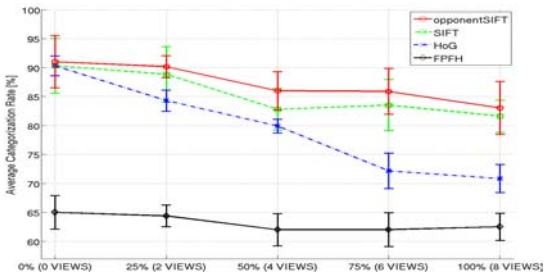


Fig. 6. Performance of descriptors under varying viewpoint. During testing, we varied a number of cases in which object is presented to the system from unknown viewpoint between 0 and 8 per object (0-100% test views).

TABLE III

RESULTS FOR THE FEATURE SELECTION EXPERIMENTS FOR *Setup-50*.

Descriptor	Av. Categ.Rate	σ
SIFT	82.8%	3.6%
opponentSIFT	**86.0%**	3.3%
HoG	79.9%	1.2%
FPFH	62.0%	2.8%

training and testing, and data for 11 objects for subsequent validations. Due to the fact that we aim to test performance of the system for the object categorization and not object instance recognition, an object that was presented to the system during the training phase was never used later to evaluate the performance.

In order to train the system, we selected 8 views per object separated by 45° (Fig. 5 top row). The system was evaluated on the same amount of data. However, to assess robustness of different object representations under varying viewpoint condition, the test set includes also data collected from the different viewpoints than the training data. We varied a number of *unknown* viewpoints between 0 and 8 per object (it gives between 0-100% of the data in the test set). We established the experimental setup in which half of the objects is presented to the system from the unknown viewpoint, *Setup-50*. It is illustrated in Figure 5. We assumed that *Setup-50* best reflects the real condition. The results are reported for a single object view and information provided by different views was not fused. To average the results each experiment was repeated five times for randomly chosen object instances. We report the average categorization rate and standard deviation (σ).

C. Feature Selection

First, we evaluated performance of different descriptors under varying viewpoint conditions. For that purpose, we built four identical single cue OCSs, one for each descriptor. We selected descriptors to provide a high recognition rate[2] and encode different object properties: appearance (SIFT), color (opponentSIFT), couture shape (HoG) and 3D shape (FPFH). A segmented part of the image was resized, such that the shorter dimension equals to 200 pixels. The SIFT and opponentSIFT were extracted using a grid detector, and HoG descriptor using the Canny edge detector. The final object representation for the 2D descriptors follows a concept of the spatial pyramid, and for the 3D descriptor BoW model for words found using the KNN clustering. The experimental setup is presented in Table II.

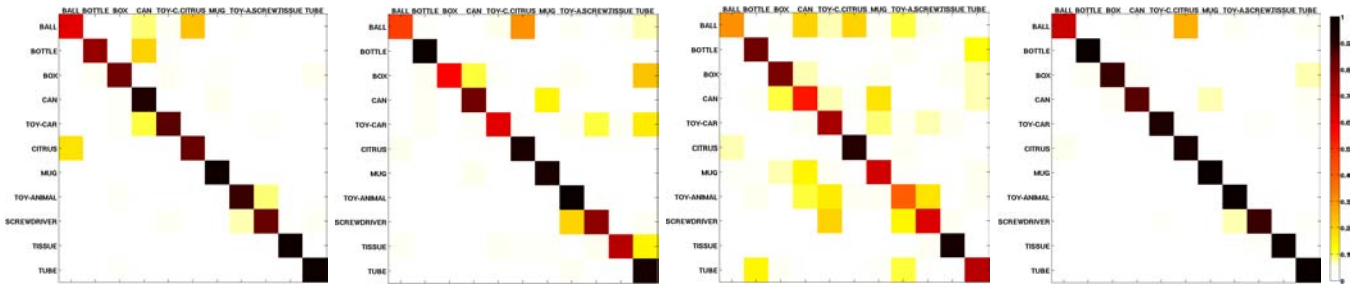
In order to assess the performance of the descriptors under different viewpoints, we varied a number of *unknown* viewpoints in the test set between 0 and 8 (see Section V-B). The results are illustrated in Figure 6. All 2D descriptors obtained rather high categorization rate when the viewpoint was known (0%), but the performance dropped significantly when as the viewpoint varies. The highest performance was obtained for the color descriptor (opponentSIFT) which naturally indicates that color information is less influenced by the viewpoint changes than shape information (HoG). The 2D descriptors yielded higher categorization rates than the 3D descriptor. Most probably, it is related to the quality of stereo data. However, the performance of the 3D descriptor is only slightly affected by the viewpoint changes. Additionally, we attach the numerical results for *Setup-50* in Table III.

The additional question that we wanted to answer is related to the diversity of descriptors. In the literature, different feature diversity measures have been studied. However, no consistent relationship between these measures and different combination methods was detected [37]. In practice, to judged complementary of the features confusion matrices are studied. Figure 7 (a-c) presents confusion matrices obtained for the color (opponentSIFT), contour shape (HoG) and 3D shape descriptor (FPFH). We observed that shape descriptors poorly discriminate between categories of similar shape such as a *mug*/*can* or *ball*/*citrus*. In such cases, the color descriptor demonstrated higher discrimination power.

D. Cue Integration

In this section, we present results from combining 2D and 3D categorization. The 2D-3D cue integration was obtained by integrating evidences from the single cue OCSs. We applied both the linear and nonlinear algebraic combination

[2]Average categorization rate for other tested descriptors [9] (*Setup-50*): rgbHistogram 36.8%, hueSIFT 59.1%, rgSIFT 83.5%, rgbSIFT 85.4%. We are planning to evaluate other very recently presented 3D descriptors: RSD [14] and VFH [16].



(a) opponentSIFT (b) HoG (c) FPFH (d) opponentSIFT+HoG+FPFH

Fig. 7. Confusion matrices obtained for: (a) color (opponentSIFT), (b) contour shape (HoG), (c) 3D shape (FPFH) descriptor, and (d) integrated opponentSIFT+HoG+FPFH (linear combination method, sum rule). The images are best viewed in color.

methods to classifier outputs and results are given in Tables IV and V.

The following pairs/triples of the single cue OCSs were combined: opponentSIFT+FPFH, opponentSIFT+HoG, HoG+FPFH, and opponentSIFT+HoG+FPFH. An integration of more than three OCSs imposes practical difficulties in estimating system parameters. With respect to the type of integrated descriptors, we observed the same trend in performance for both the linear and nonlinear combination methods. The best categorization rate was obtained for fusion of all three descriptors (opponentSIFT+HoG+FPFH). The second best for the combination of descriptors that capture different object attributes and originate from different channels (2D and 3D): 2D color and 3D shape descriptor (opponentSIFT+FPFH). Then, the third performance was obtained for the color and shape descriptor originating from the same channel (opponentSIFT+HoG), and the lowest for the two shape descriptors (HoG+FPFH).

In case of the linear algebraic methods, we tested the wighted sum, product and max rule. For all combinations of features, the approach based on the sum and product rule improved the performance of the system in comparison to the best single cue OCS (based on opponentSIFT), and the sum rule was superior to the product rule. The max rule that in case of two classifiers is equivalent to the majority voting, yielded the lowest categorization rate. In case of the nonlinear algebraic methods (integration based on SVMs), we evaluated the RBF, χ^2 and histogram integration functions. The difference in performance between nonlinear functions is best visible for combination of all three descriptors, and χ^2 function yielded the highest categorization rate.

Finally, the overall best performance of **92%** was obtained for integration of the three descriptors using the linear combination method. When comparing to the best single cue OCS (based on opponentSIFT), the combination of 2D and 3D features improved performance of the system by **6%**. The final confusion matrix obtained for this experiment is presented in Figure 7 (d). When comparing it to the confusion matrices for the single cue OCSs, it is clearly visible that cue integration significantly improved performance for all the classes and the most difficult remained differentiation between the *ball* and *citrus* category. In our study, the linear algebraic integration methods outperformed the nonlinear methods. It is most probably due to the fact that a small set of data was used to train the SVM classifier for the nonlinear methods. We can draw the conclusion that in case of a limited amount of data, the simpler fusion methods are more efficient.

E. Task-constrained Grasping

In this section, we show the results of an integrated system considering categorization for task-constrained object grasping. Our experimental scenario is to plan grasps on multiple objects constrained by the assigned tasks, taking into account robot's embodiment. The robot is presented with a scene containg several unknow objects and an example of such a scenario is presented in Figure 8. The objects are segmented from the background in a real time (as described in Section III-A) and our 2D-3D object categorization system is applied to each generated object hypothesis. In the given scene, four objects are found and from left to right they are correctly classified as a mug, screwdriver, bottle and mug. Figure 9 presentes the classification confidence values obtained in this expariment for the four object hypothesis. In the future work, we plan to use the confidence values to further improve generation of grasping points.

Next, given the assigned task, the robot need to decide (1) which object should be grasped, and (2) how to grasp it to fulfill the task requirements. For this purpose, we use the embodiment-specific task constraint model. The model is prior trained on the grasp database that includes the stable grasps generated on a set of 48 objects using the hand model of the humanoid robot Armar [38]. The 48 objects are from the Princeton Shape Benchmark [39]. They cover the 6 object classes (8 models per class), each of which includes 4 different object shapes scaled to 2 sizes – small and average. Three tasks were labeled: *hand-over, pouring* and *tool-use*. The total training set include 1227 cases with 409 cases

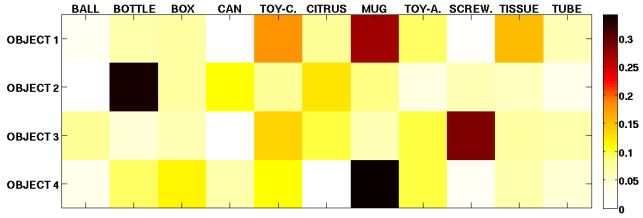
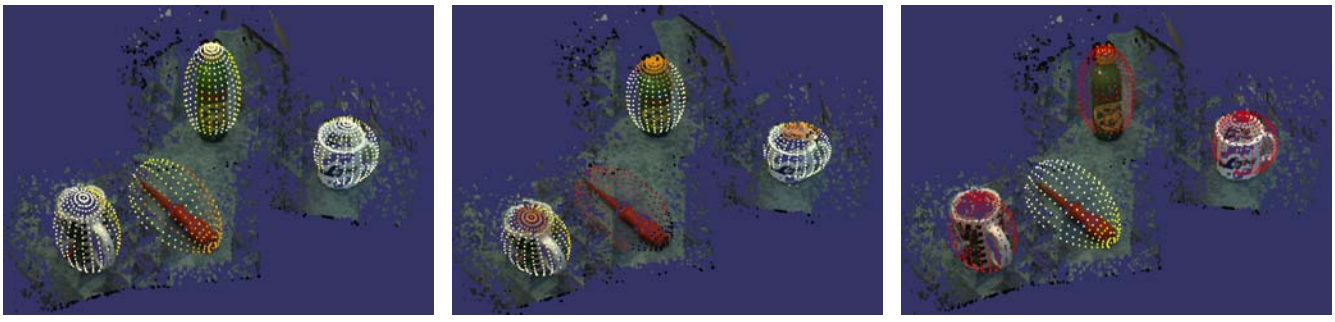


Fig. 9. Classification confidence values obtained for the four objects in the real scene presented in Fig. 8. The final classification decision is made by choosing the class with the strongest support. For each object, the confidance values are normalized to sum to one over all classes.



(a) Hand-over (b) Pouring (c) Tool-use

Fig. 8. Generated grasp hypotheses and associated probabilities for three different tasks: (a) hand-over, (b) pouring and (c) tool-use, and four objects classified as: a mug, screwdriver, bottle and mug (from left to right). The grasping probability around an object is indicated by color of a point and the legend is presented in (d) (the brighter is the point the higher is the probability). The images are best viewed in color. For the accurate 3D information, we kindly direct the reader to our web site `http://www.csc.kth.se/~madry/research/task_constrained_grasping/index.php` where the movies with the experimental results are available.

TABLE IV

RESULTS FOR THE CUE INTEGRATION USING LINEAR ALGEBRAIC COMBINATION METHODS.

Descriptors D1+D2(+D3)	Max Rule			Product Rule			Sum Rule		
	Average Categ.Rate	σ	Av. Gain D1,D2,(D3)	Average Categ.Rate	σ	Av. Gain D1,D2,(D3)	Average Categ.Rate	σ	Av. Gain D1,D2,(D3)
opponentSIFT+HoG+FPFH	84.5%	5.8%	-1.5,4.6,22.4%	89.9%	4.9%	3.9,10.0,27.9%	**92.0%**	2.8%	**6.0**,12.0,29.9%
opponentSIFT+FPFH	86.6%	3.7%	0.6%, 24.5%	87.8%	4.3%	1.9%, 25.9%	90.9%	3.2%	4.9%, 28.9%
opponentSIFT+HoG	81.9%	3.4%	-4.1%, 2.0%	86.0%	0.8%	0%, 6.1%	87.4%	0.6%	1.4%, 7.5%
HoG+FPFH	79.9%	1.5%	0.0%, 17.9%	83.1%	6.3%	3.1%, 21.0%	83.4%	4.6%	3.5%, 21.4%

TABLE V

RESULTS FOR THE CUE INTEGRATION USING NONLINEAR ALGEBRAIC COMBINATION METHODS (SVM-BASED CUE INTEGRATION).

Descriptors D1+D2(+D3)	RBF Kernel			χ^2 Kernel			Hist. Intersection Kernel		
	Average Categ.Rate	σ	Av. Gain D1,D2,(D3)	Average Categ.Rate	σ	Av. Gain D1,D2,(D3)	Average Categ.Rate	σ	Av. Gain D1,D2,(D3)
opponentSIFT+HoG+FPFH	87.6%	6.5%	1.6,7.7,25.6%	90.4%	0.3%	4.4,10.5,28.4%	89.8%	1.8%	3.8,9.9,27.8%
opponentSIFT+FPFH	87.2%	2.0%	1.2%, 25.2%	87.6%	1.6%	1.6%, 25.6%	87.2%	2.8%	1.2%, 25.2%
opponentSIFT+HoG	84.3%	5.4%	-1.7%, 4.3%	84.8%	5.3%	-1.1%, 4.9%	84.7%	3.1%	-1.3%, 4.7%
HoG+FPFH	79.1%	1.8%	-0.8%, 17.0%	79.5%	3.0%	-0.4%, 17.5%	80.4%	1.2%	0.5%, 18.4%

per grasping task. Figure 4 shows the learned structure of the task constraint Bayesian network. Table I describes each feature represented by the network. Note that, compared to other object features $size$, $cvex$ and $ecce$, object class $obcl$ is directly conditioned by the $task$ node. This is consistent with our previous results on human-hand specific network [3]. This result confirmed that the object category information is the most important object feature that represents its task affordances.

Note that the BN allows us to infer local distribution on each individual or small set of variables, based on partial or complete observation of others. In this work we are interested to infer, in the clustered 3D environment on the table top shown in Figure 8, the most suitable grasp position pos given the assigned task $task$ and the categories of the objects $obcl$. This can be demonstrated by a likelihood map on a set of grasp position pos around each object, i.e. $P(pos|task, obcl)$. The point that has the highest P (indicated by the brightest color) would imply the best grasp position for the task.

To do so, we first sampled a set of grasping points pos on an ellipsoid around each object in the scene that is visible by the camera (see Figure 8). Since pos in the Bayesian network is represented in the object local coordinate, a neccessary step is to convert the pos data sampled in the world corrdinate frame to the object coordinate system. This transformation requires the knowledge of object position and orienation in the world coordinate. In this paper, we assume that the orientation of the object is known, and the position of the object is estimated by fitting the synthetic object model with the same class to the point cloud of the real object.

Figure 8 shows the results of the experiment. From left to right, we show the likelihood maps using colored sample points of $P(pos|task, obcl)$, when task is *hand-over* (a), *pouring* (b) and *tool-use* (c) respectively. We see that for the *pouring* task, the likelihood of the sample points around the screwdriver is clearly lower than the other three objects, indicating the screwdriver can not afford *pouring* task. While the observation is opposite for the *tool-use*. And for the *hand-over* task, all the four objects have high likelihood. This indicate that using the object category information and the task constraint BN, we can successfully select the object according to their task affordance.

For the object that affords the assigned task, for example the bottle and the two mugs in Figure 8 (b), the likelihood maps show darker color on the top of the object. This is because to pour the liquid, we should not grasp from the

Preprint submitted to 2011 IEEE/RSJ International Conference
on Intelligent Robots and Systems. Received March 28, 2011.

top and block the opening of the object. To *hand-over* the screwdriver, in Figure 8 (a), we see that the network favors the position around the tip of the screwdriver whereas leaving the handle part for regrasp.

VI. CONCLUSIONS AND FUTURE WORK

Robots grasping objects in unstructured environments need the ability to select grasps for unknown objects and transfer this object between objects based on their categories and functionalities. Altough object categorization is one step toward this goal, it does not solve all the problems. The first problem is the ability of a robot to perform this in real scenes, thus generating object hypotheses for unknown objects and also in 3D. Although for pure categorization 2D information may be suffiecient, 3D information is neccessary if grasping and manipulation of objects is the final goal.

We have presented a 2D-3D object categorization system that is built upon an active scene segmentation module. The system allows generating object hypotheses and segmenting them from the background in real-time. Results from the extensive experiments in the real environment showed that the proposed system achieved high object recognition rate (up to 92%), significantly better than the classic single cue SVM in the same task. Moreover, the simple cue integration method proposed in this paper is much more efficient and effective especially in the situations where limited ammount of data is available.

The categorization system is integrated with a task constrained model for goal-directed grasp planning. We showed that the object category information can be efficiently used to infer the task affordance of the observed objects. And the integrated system allows reasoning and planning of goal-directed grasps in the real-world scenes with multiple objects.

One avenue for the future research is the integration of the proposed system with the on-line stability estimation system proposed in [40]. The aim will be to condition the choice of grasps based on the perceptions available to a robot prior to and while lifting and transporting an object.

REFERENCES

[1] M. Bjorkman and D. Kragic, "Active 3D scene segmentation and detection of unknown objects," in *ICRA*, May 2010.

[2] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning Task Constraints for Robot Grasping using Graphical Models," in *IROS*, 2010.

[3] D. Song, C.-H. Ek, K. Huebner, and D. Kragic, "Multivariate discretization for bayesian network structure learning in robot grasping," in *ICRA*, May 2011, to appear.

[4] J. G. Greeno, "Gibson's Affordances," *Psychological Review*, vol. 101, no. 2, pp. 336–342, 1994.

[5] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, "Contact-reactive grasping of objects with partial shape information," in *IROS*, 2010.

[6] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann, "Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot," in *ICRA*, 2010.

[7] G. D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 91–110, 2004.

[8] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR*, 2008, pp. 1–8.

[9] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1, 2005, pp. 886–893.

[11] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *CVPR*, 2003, pp. 409–415.

[12] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," in *SMI*, 2004, pp. 145–156.

[13] A. Johnson, "Spin-images: A representation for 3-D surface matching," Ph.D. dissertation, Carnegie Mellon University, 1997.

[14] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3D modelling of novel objects from a single view," in *IROS*, 2010.

[15] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *ICRA*, May 2009.

[16] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *IROS*, 2010.

[17] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision (ECCV)*, 2004, pp. 1–22.

[18] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, vol. 2, 2006.

[19] M. Weber, M. Welling, and P. Perona, "Towards automatic discovery of object categories," in *CVPR*, vol. 2, 2000, pp. 101–108.

[20] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *CVPR*, vol. 2, 2003, pp. 264–271.

[21] M. Sun, H. Su, S. Savarese, and L. Fei-Fei, "A multi-view probabilistic model for 3d object classes," in *CVPR*, 2009.

[22] J. Matas, R. Marik, and J. Kittler, "On representation and matching of multi-coloured objects," in *ICCV*, 1995, p. 726.

[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley, 2001.

[24] R. Polikar, "Ensemble based systems in decision making," *Circuits And Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[25] M. E. Nilsback and B. Caputo, "Cue integration through discriminative accumulation," in *CVPR*, 2004.

[26] A. Pronobis, O. M. Mozos, and B. Caputo, "SVM-based discriminative accumulation scheme for place recognition," in *ICRA*, 2008.

[27] D. Marton, Z-C.and Pangercic, R. B. Rusu, A. Holzbach, and M. Beetz, "Hierarchical object geometric categorization and appearance classification for mobile manipulation," in *Humanoids*, 2010.

[28] R. Sara, "Finding the largest unambiguous component of stereo matching," in *ECCV*, vol. 2, May 2002.

[29] *ROS Point Cloud Library*, http://www.ros.org/wiki/pcl, Last visited: Feb 2011.

[30] J. C. Platt, "Probabilistic outputs for SVMs and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, pp. 61–74, 1999.

[31] A. Pronobis and B. Caputo, "Confidence-based cue integration for visual place recognition," in *IROS*, 2007, pp. 2394–2401.

[32] K. Huebner, "BADGr - A Toolbox for Box-based Approximation, Decomposition and GRasping," in *Workshop on Grasp Planning and Task Learning by Imitation (IROS)*, 2010.

[33] A. T. Miller and P. K. Allen, "GraspIt! A Versatile Simulator for Robotic Grasping," *IEEE Robotics and Automation Magazine*, 2004.

[34] *Caltech101 Database*, Last visited: Feb 2011, http://www.vision.caltech.edu/Image_Datasets/Caltech101.

[35] *Princeton Shape Benchmark*, http://shape.cs.princeton.edu/benchmark, Last visited: Feb 2011.

[36] *KIT ObjectModels Web Database*, Last visited: Feb 2011, http://i61p109.itec.uni-karlsruhe.de/ObjectModelsWebUI/.

[37] C. A. Shipp and L. I. Kuncheva, "Relationships between combination methods and measures of diversity in combining classifiers," *Information Fusion*, vol. 3, pp. 135–148, 2002.

[38] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *Humanoids*, 2006.

[39] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *SMI*, 2004, pp. 167–178.

[40] Y. Bekiroglu, K. Huebner, and D. Kragic, "Integrating grasp planning with online stability assessment using tactile sensing," in *ICRA*, 2011.