A Large-Scale Multi-Pose 3D-RGB Object Database

Fabian Sachara*, Finn Handmann*, Nico Cremer*, Thomas Kopinski†, Alexander Gepperth‡, Uwe Handmann*
*University of Applied Sciences, Hochschule Ruhr West, Lützowstrasse 5, 46236 Bottrop, Germany
firstname.lastname@institut-informatik.de
†Fachhochschule Südwestfalen, Lindenstraße 52, 59872 Meschede, Germany
kopinski.thomas@fh-swf.de
‡Applied Computer Science, University of Applied Sciences Fulda, Germany
alexander.gepperth@cs.hs-fulda.de

***Abstract*—We present a new RGB-D database for multi-pose object recognition tasks. With the help of a multi-axis rotation framework, we are capable of capturing depth and color data of arbitrary small objects from virtually any viewpoint. In addition, recording is performed in a nearly lossless fashion, avoiding typical bleeding artifacts present in related reference data bases. This contribution presents the main advantages of our setup and contrasts it against other reference data bases. Furthermore, it outlines possible use cases and application scenarios of our data set and is complemented by experiments with standard machine learning techniques used in, e.g., object recognition tasks within the robotics domain. The experiments demonstrate the validity of our data base as they corroborate that viewpoint variance is indeed an important factor to take into account for object detection, which, from our perspective, is sometimes not considered at the required level. Detection accuracy is high if samples can be trained on data taking into account as many viewpoints as possible.**

I. INTRODUCTION AND RELATED WORK

When dealing with the problem of instance-based object recognition, one has to simultaneously take into consideration various factors on many different levels such as sensor technology/quality, algorithm design and, last but not least, environmental factors. The latter comprise difficulties arising from illumination, object occlusion or, within the domain of robotics, difficulties resulting from data acquisition by an autonomous agent. In such a scenario, one possible desired goal for a robot is to find and precisely locate instances of a target or known item, for example in order to pick it up or to process the information further. However, it is not always a simple task to achieve, due to occlusions, imperfect data or even multiple instances appearing in the scene. In order to separate the impact of each of these factors from the performance of learning algorithms for object recognition, reference databases containing near-perfect data are required. Such databases allow to benchmark a given learning algorithms separately from any preprocessing steps that may be necessary to attenuate environmental factors. As a consequence, the availability of reference databases has grown in recent years, keeping track with advances in sensing technology. With the advent of accurate and cheap RGB-D sensing technology, the construction of reference databases for object recognition from RGB-D data has become a major requirement for advances in RGB-D object recognition. However, existing databases typically share the drawback of lack of diversity with respect to object viewpoints, and moreover suffer in terms of data quality due to the way in which the data is acquired. Typical solutions to these problems include a multi-sensor setup to induce greater viewpoint variance, as well as data post-processing algorithms which can alleviate the flaws in data recording.

In this contribution, we present a data recording design overcoming these flaws simultaneously by its simple yet effective setup, allowing for fully automated object data acquisition resulting in superior data quality. The design itself consists of a rotational arm capable of securing target objects by nylon wire while being able to fully rotate around two different axes. Data is recorded with a single camera at a fixed position in near-range taking color and depth snapshots at equidistant time-intervals. In this way, virtually any small object can be recorded from any viewpoint without any intermediate human intervention, leading to almost perfect data snapshots (no segmentation step necessary to separate object from background) and a high number of possible objects.

This contribution is laid out as follows: In Section II begin by describing the most important work related to the field of object recognition from data-driven approaches based on reference databases available online. Subsequently, the specific contribution of the recording setup along with its main advantages presented in this paper is described in Section III. The details and and system overview are then provided in Section IV. Section V presents the means by which data is acquired along with the most important technical specifications. Subsequently, the resulting database is described in detail in Section VI. In order to get a grasp on how to use this dataset as a reference, benchmark tests have been conducted and the most important findings are presented in Section VII. Conclusively, we provide a summary of this contribution in Section VIII along with an outlook on possible applications and future work.

II. RELATED WORK

Databases typically aim at providing a large number of instances organized within a certain taxonomy in order to maintain the diversity necessary for algorithm development and testing. To this end, usually an automated system is set up in such a manner that a single object can be placed into a certain scene for recording and subsequent post-processing of data. Here lie some of the potential pitfalls, as the algorithms developed for this post-processing are designed to perform some form of object-background cropping which is a non-trivial task in nearly all cases, and rather a research topic of its own. The Washington RGB-D dataset consists of 300+ common household objects organized into 51 categories [1]. Samples were recorded with a Kinect-style camera with the objects

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

placed on a turntable. To induce viewpoint variance the camera was moved to different heights for each of the three recordings per object. As can be seen from the samples included in the datasets, due to imperfect object-background cropping, parts of the objects are missing in nearly all cases. This is undesirable as incomplete samples do not reflect the real-world scenario well and moreover are prone to errors. The viewpoint variance realized by the camera shift is limited to three different angles and does not provide an exhaustive description of the included samples in terms of both depth and RGB data. The BigBIRD database contains data samples from 125 different objects at this time of writing [2]. RGB data is present in the form of 12 megapixel images while depth data is available in the form of point clouds recorded with a Carmine 1.09 sensor. The number of instances is large, albeit much inferior to the aforementioned set. Its main benefit are the high resolution images and the added viewpoint variance which mainly comes from setting up five different cameras, which however remain in a fixed position. This allows for an easy and precise mesh reconstruction of objects, but still does not densely cover the viewpoint angles of an object. Furthermore, it suffers from the ”bleeding” of the surrounding environment during recording. This is mostly resolved by a post-processing step, however significant artifacts remain attached to the object's data points. Lastly, camera calibration is necessary for this setup in order to achieve precise measurements. The interference between IR patterns is avoided by time-multiplexing, i.e. turning cameras on one at a time. In order to increase viewpoint variance for the specific task of hand gesture recognition the authors of [3] propose the variation of the objects, i.e. hands, themselves. This is a feasible approach for this specific task as, during the recording of the database, the participants are instructed to translate and rotate the hand to induce as much variance as possible. This is, however unfeasible if object cropping has to be taken into account by the post-processing algorithms which typically leads to imperfect data samples. Other approaches, such as MOPED [4] provide high quality images of a large number of object instances. However, while facilitating high-res cameras is a valid approach, it is difficult to perform comparisons between systems making use of commodity hardware available in standard RGB-D sensors. The Willow Garage Dataset provides 160 frames of household objects taken recorded with a first-generation Kinect and providing 6DOF poses for each object. Along with the aforementioned problems of object segmentation and minor viewpoint variance, cumbersome checkerboard calibration has to be implemented increasing the risk for potential errors. Typical (other) drawbacks therefore are calibration difficulties, few object samples, low data quality, minor viewpoint variance or low-resolution data [5], [6], [7].

III. Contribution and Novelty

In this contribution, a novel data recording technique is proposed for establishing a database aimed at instance recognition from RGB-D data. The unique characteristic of our approach is its setup, as it allows for a viewpoint variance at a scale not present in current publicly available data sets. The framework permits recording from any desired angle, with possible angle step sizes as small as 4 degrees. This is more than sufficient to achieve multi-pose recordings for common object recognition problems or mesh reconstruction techniques.

Furthermore, a depth segmentation technique is implemented allowing for almost lossless object recording via volume of interest (VOI) cropping. Thereby, various difficulties are avoided such as bleeding, environmental reflections or other artifacts, resulting from, e.g., imperfect segmentation of objects standing on a turntable. The sensor coming to use is pre-calibrated for RGB and depth data, therefore color and depth data points can be mapped to a single resulting RGB-D point cloud.

IV. System overview

The system operates on a revolving plate onto which the rotating frame is set up (cf. Figure 1). Each object is chucked with nylon cords into the frame's rotating arm. This arm is mounted to the main frame's vertical axis allowing to perform full 360 degree rotations for the objects by varying the polar angles. A stepper motor, attached to the bottom part of the main frame's axis, controls the arm's step-wise rotation. The revolving rotation of the platform is realized via a second stepper motor, which is mounted to the side of the platform. The cogwheel attached to the stepper motor bites into the large cogwheel (on the floor) which in turn is fixed around the main platform. This way, precise steps can be performed by the system with a precision of up to 4 degrees in azimuthal angle steps.

As this setup needs to freely rotate in both angular directions, an own power source is placed onto the rotating platform and moreover a wireless connection needs to be established to take and transmit snapshots after each rotation step. The stepper motor responsible for the rotation of the platform is connected and controlled by an Arduino microcontroller. This, in turn, is connected via USB to a Raspberry Pi computer responsible for the system control on the platform and maintaining the Wifi connection to the camera system, which is responsible for the recording. The power source on the platform allows for an operating time of several hours. During a recording session, each object is fixed in place by three nylon cords which results in a stable positioning of the object during the rotation phase. Slight vibrations from the arm rotation are partially passed on to the object, however do not affect the quality of the recording as a fixed 'pre-recording delay' can be defined in which vibration can die down. Moreover the nylon cords are thin and translucent hence do not appear in the resulting snapshots. The camera is positioned at approximately the same height as the target object's centroid and remains fixed for the setup.

V. Data recording

Data is recorded via the Creative Gesture Camera (cf. Figure 2) from a fixed position. Each snapshot results in a 640x480 RGB image and a 320x160 depth image recorded by its Time-of-Flight (ToF) sensor, also denoted point cloud (PC).

For each object, we assume its centroid being at a fixed position and fixed distance in the area in front of the camera. A VOI is assumed in between the frame's arms and all data points contained within the VOI are extracted and saved. The difficulty is to extract the valid data points from the whole point cloud. For certain angles, the frame's arm moves in between the camera and the target object, thereby obstructing

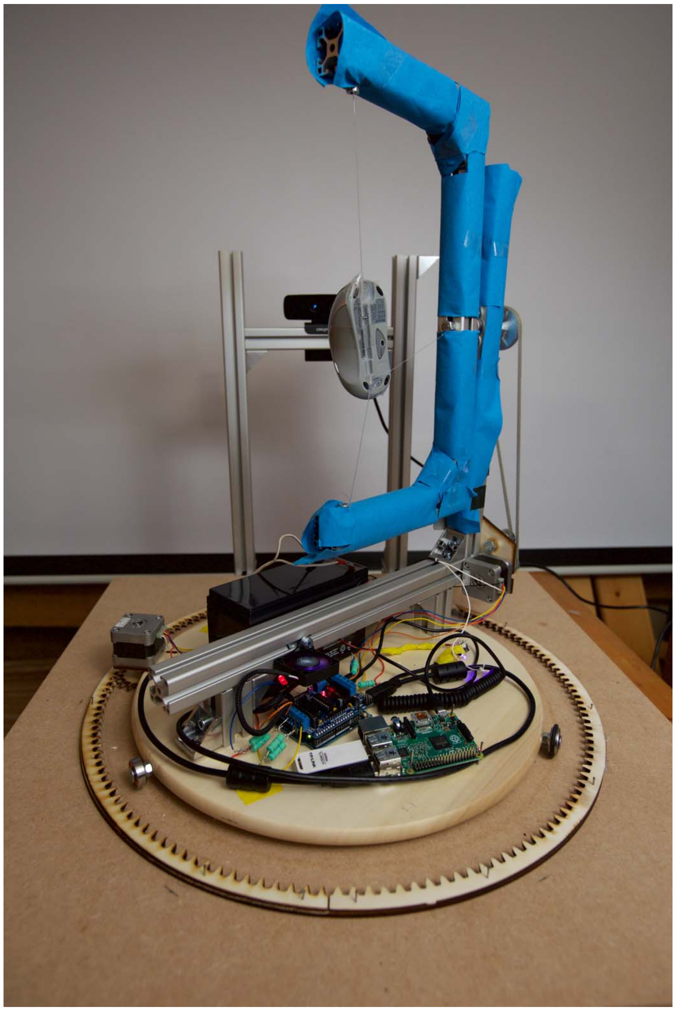

Fig. 1: The current system setup: Objects are spanned into an arm able to perform full 360°rotations. The whole construction is driven by small stepper motors allowing for the arms to shift in 4°angle steps. An own power source is placed onto the rotating wheel while the an RGB-D camera (in the background) takes snapshots after each step.

the recording. These snapshots can easily be detected by a distance thresholding and are subsequently removed from the database. Additionally, the arm is coated with a sheet of distinct color allowing for refined differentiation between valid and non-valid (i.e. artifacts, outliers) data points. The more difficult cases are reflections caused by the frame itself in those cases in which the rotation of the arm caused its own position to possibly be within the VOI. This can be circumvented on the one hand through simple color cropping or, for the more difficult cases, by a specifically designed tree-cropping algorithm.

The main idea of tree-cropping algorithm is to obtain all data points belonging to the object, while removing all outliers and artifacts. The data is cropped using a tree originating from the suspected (0, 0, 0) reference point of the object's origin. Paths are established by propagating from the existing leafs of

Fig. 2: The Creative Gesture Camera, capable of recording RGB images at a resolution of 640×480 and depth images at a resolution of 320×160 at a frame rate of 30 Hz.

the tree to the nearest neighbors. All points which cannot be connected through a path fulfilling the necessary criteria are cropped. The main criteria consist of a minimum threshold for the individual distance between 2 points in the tree, Θ^2_{min}, and a threshold for average distance between n-points of the tree Θ^n_{avg}. In order to maintain the database at a manageable size, we initially configure the angle steps in such a way as to produce roughly 100 snapshots per object: Snapshots are taken after 36 degree rotations for both the polar and the azimuthal angles, i.e. the arm and the platform each perform 10-step rotations. More precisely, the arm performs a full 3D rotation of the object in polar angle direction in ten steps, after which the platform rotates 36 degrees in azimuthal angle direction (also 10 times in total). This yields 100 sample snapshots for every object after which the cropping algorithm reduces the set to contain only valid samples (i.e. removing those with too many artifacts).

VI. The Database

The database consists of 60+ sample objects and will be constantly extended. It mainly contains objects from indoor environments, grouped taxonomically with a special focus on the usability for robotic applications. Objects having roughly the same shape and color characteristics from all angles (e.g. a ball) will also be included, however with only few reference samples, as they do not provide a large data variety. At the current state of recording, three taxonomic groups are included in the database: toys, kitchen objects and miscellaneous. The latter contains objects which cannot be properly assigned to any group, however may well be present in a household scenario. Due to the need to maintain the stability of the setup, there is a natural weight and size limit for the recording of an object.

Figure 4 shows a sample recording of a toy bridge in ten different azimuthal angles ϕ, completing a full rotation around the z-axis. Each image is color coded with green data points being closer to the camera setup and orange/red points being further away. Depending on the angle of the object towards the camera, more or less noise is visible per snapshot, as e.g. in the top row in the third and fourth sample from the left. This is

Fig. 3: Sample set of household objects taken from the kitchen set (top row) and the miscellaneous set (bottom row).

the result of a depth recording with a ToF sensor as less light is reflected back directly towards the camera, and therefore measurements become less precise. Moreover, the quality of ToF depth measurements depends on the object's distance from the sensor - the further away the object is, the less precise the measurements can be. Another factor having a possibly strong impact on the quality is an object's reflection coefficient. This coefficient defines the amount of light reflected from an object back towards the light source (in this case the sensor's NIR light emitter), hence a low coefficient may be responsible for imprecise measurements.

The rightmost sample in the top row is an unrecoverable artifact due to either to much noise stemming from the frame, or the frame itself being (partially) in the field of view (FOV) of the camera. All snapshots are saved and provided as .pcd files containing both the depth and the color information. Additionally we provide the raw scenery data as well as the RGB and color coded depth files separately. This allows for easy integration into the well-known and established point cloud library (PCL) [8]. The advantage of our approach is that we provide high-quality RGB-D data, recorded with up-to-date sensing hardware, from virtually any desired viewpoint, allowing to analyze and benchmark state-of-the-art machine learning and computer vision approaches in a systematic fashion. The main drawback of other approaches is contained in the cropping process and the limitation of angular perspectives which our system alleviates in many aspects. The combination of VOI depth cropping and the aforementioned tree-cropping algorithm yields very satisfactory point clouds along with the RGB data also containing the noise stemming from a typical sensor recording. No other data cropping scheme is required other than those described in this section yielding improved results in terms of data quality as other objects in comparable reference databases require either e.g. further post-processing or object-background seqmentation techniques. Other approaches setting up similar databases usually struggle with the cropping process, as each object is placed on some kind of platform which results in the need to segment it from the underlying plane. This subsequently makes it difficult to find precise cropping techniques and yields imperfect shape/color structures with potentially misaligned points. Moreover, the surrounding environment always reflects a certain amount of light onto the object, merging and therefore corrupting the reference objects. This is not the case for our setup as we make sure each object is attached freely to the frame with a minimum distance to any surrounding nearby object.

In addition, we approximately overcome the problem of angle restriction, prevalent in some form in all other databases. The usual approach is to provide a variation by repositioning the camera in a certain angle or by setting up several cameras at once. Either way, the limitations are severe as the number of angles is significantly smaller and moreover potentially cumbersome to vary. Any manual change in camera angle is furthermore prone to de-calibration errors which is not the case for the automated setup presented in this contribution. This is significant as one cannot rely on algorithms working robustly, since objects in a real-world scenario cannot be expected to be found in a certain angular arrangement, but are rather scattered in manifold ways. Our database provides the possibility to robustly verify such algorithms for e.g. robotics applications. Obviously, our approach has limitations as well, most notably the fact that most but not all viewpoints are possible, due to the setup shown in Figure 1 where the arm can appear between object and camera although this happens very rarely as the arm is rather slim. Furthermore, an object must lend itself to being attached by nylon wire, which is however the case for the vast majority of household objects.

Figure 5 analogously shows the RGB images for a toy horse available in our database, along with the sensor noise and the framework artifacts not yet removed (top row, sample 4 + 5) and displayed for sake of completeness. Aside from the noise stemming from the frame itself, which will be removed after tree-cropping, sensor noise will naturally remain in the sample recording itself. Due to the fact that the employed sensor is already calibrated, any further calibration is unnecessary for our setup.

VII. Benchmark Experiments

We conduct some experiments using a simple multilayer perceptron (MLP) with three hidden layers to analyze the object samples in our data base. To this end, 10 objects are selected from the database and are denoted as follows: 0 - toy bridge, 1 - toy cow, 2 - toy cube, 3 - toy basket, 4 - misc box, 5 - toy duck, 6 - misc cylinder, 7 - toy horse, 8 - toy rabbit, 9 - misc spoon. Each of these classes contains several different instances of the same class. We employ a 3D shape descriptor

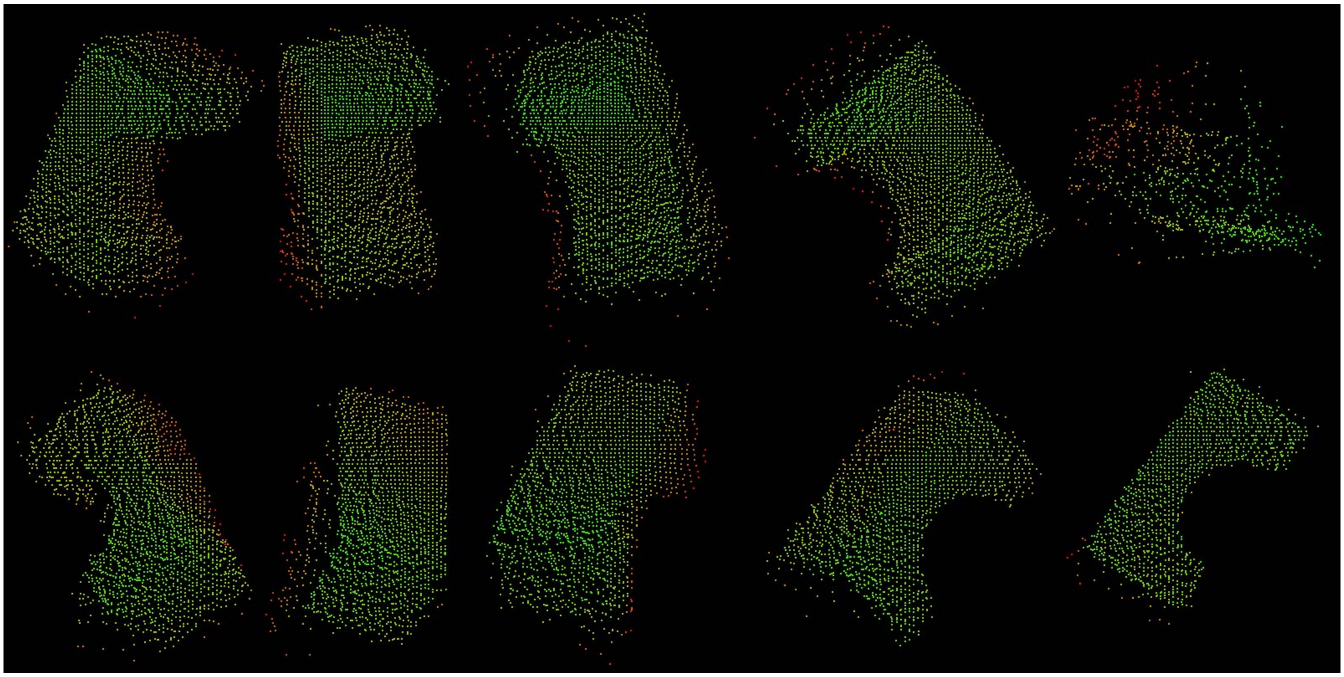

Fig. 4: Sample rotation of a toy bridge in ten different azimuthal angles. Depending on the angle of the object more (Snapshot 3, top row) or less (Snapshot 1, top row) are present. These outliers are imprecise measurements as the quality of ToF data heavily depends on the distance of the object as well as its relfection coefficient. The topmost sample represents an artifact unavoidable as the rotating arm interferes with the object's VOI during data acquisition. This sample will be removed and is not present in the database.

introduced in [9] which has yielded stable and robust results for several 3D shape recognition tasks so far. It is based on the 3D shape descriptor presented in [10] which, at its core, describes the relationship of 2 points in a point cloud by their respective tilt, yaw and pan angles. The functionality of the shape descriptor is not the core aspect of this contribution, therefore we point to the mentioned publication for further in-depth questions. The size of the input layer of the MLP corresponds to the size of the resulting descriptor (here: dim = 625), the number of hidden neurons remains at 70 (numbers empirically selected) and the number of neurons in the output class corresponds to the number of instances (here n = 10). In the classification phase, the neuron with the highest activation corresponds to the classified class. Training is conducted using the FANN library[11] with standard parameters and sigmoid activation functions. The training algorithm is RProp.

Table I shows the results as a confusion matrix of a trained MLP with 70% of the data samples used for training and 30% of the data samples left for validation. As we have 100 raw data samples per object in our data base, removing those samples which contain artifacts or parts of the frame itself results in less than 30 samples for testing. Each entry (i, i) in the matrix represents the number of correctly classified samples for this object while each entry (i, j) (with i unequal j) in every row represents the number of samples which were mistakenly classified as class j. The overall classification rate is at around 74% which is satisfactory for an unoptimized model with respect to its parameters for a problem of 10 classes. It can be seen that objects with similar shape are frequently mistaken for each other, e.g. row 2 where the toy cube is mistaken for the toy bridge on 6 occasions. Contrary the spoon (class 9) is classified correctly in all of the cases (100% recognition rate).

	bdg	cow	cube	bskt	box	duck	cyl	hrs	rbt	spn
bdg	11	0	3	0	4	1	3	1	1	0
cow	1	16	1	0	0	0	0	5	1	0
cube	6	0	17	0	0	0	0	0	1	0
bskt	0	1	1	20	0	0	0	0	0	0
box	2	0	2	0	20	0	0	0	0	0
duck	1	2	1	0	1	16	0	1	2	0
cyl	0	0	0	0	0	0	23	1	0	0
hrs	2	3	0	0	2	1	0	13	3	0
rbt	0	1	1	0	0	2	1	2	17	0
spn	0	0	0	0	0	0	0	0	0	24

TABLE I: Confusion Matrix for MLP testing with 70 hidden neurons and an overall accuracy of 74% (abbrv. for convenience)

	bdg	cow	cube	bskt	box	duck	cyl	hrs	rbt	spn
bdg	12	2	2	1	1	2	4	0	3	3
cow	2	16	0	0	0	3	2	4	2	1
cube	6	2	19	0	1	0	0	2	0	0
bskt	0	2	1	22	0	2	0	0	0	0
box	3	1	0	0	22	0	0	1	2	1
duck	1	0	1	1	1	19	2	1	3	1
cyl	1	0	1	0	0	0	27	0	0	0
hrs	3	9	0	0	1	1	0	14	2	0
rbt	2	3	0	0	0	0	1	3	19	2
spn	0	0	0	0	0	2	0	0	0	28

TABLE II: Confusion Matrix for MLP testing with 70 hidden neurons and an overall accuracy of 66% (with artifacts) (hidden=70) (abbrv. for convenience)

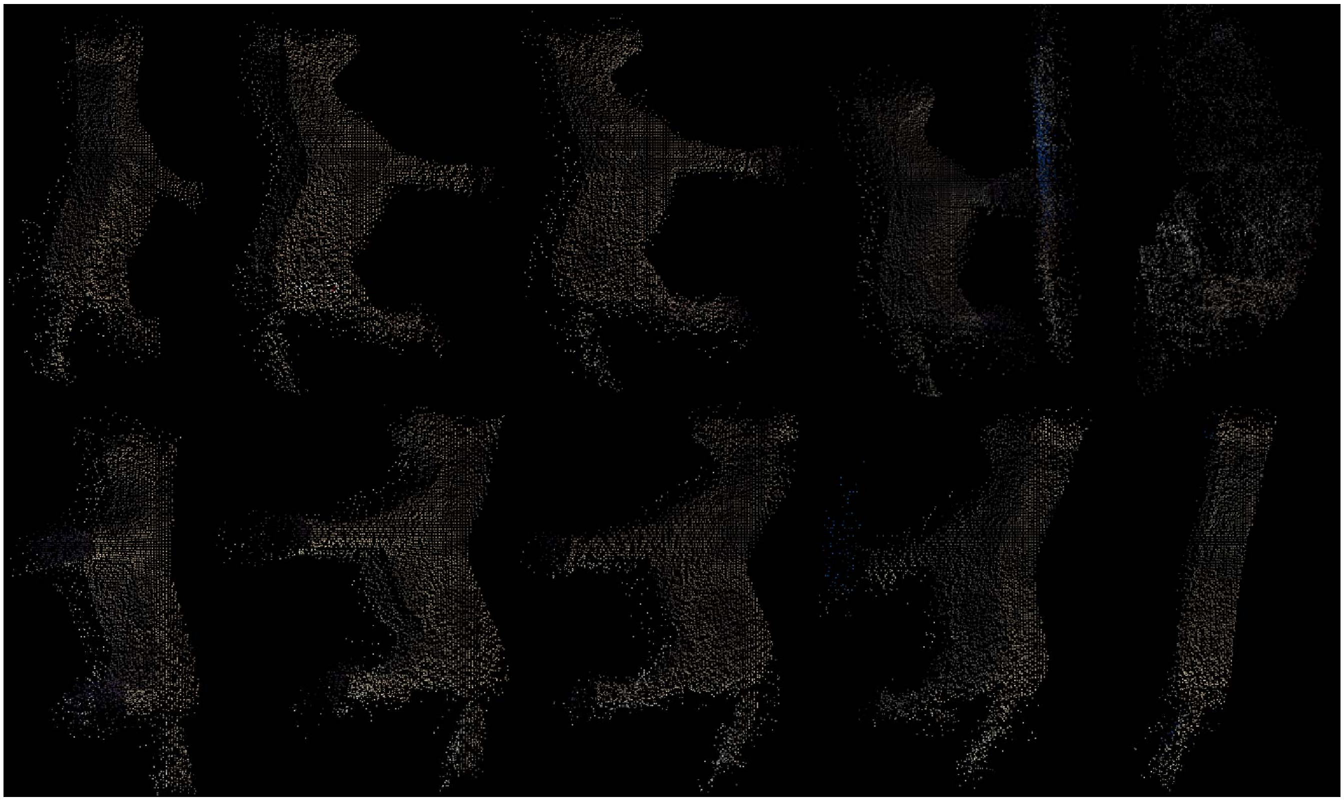

Fig. 5: RGB snapshots of a toy horse in our database viewed from ten different azimuthal angles. Note the artifacts before applying the cropping algorithm on the raw data (upper row, fourth sample). Again the fifth sample in the top row represents an irrecontructable sample which will be removed from the database.

Consequently, as more samples remain in the training and testing set, the classification rate drops to 66% as indicated by Table II. The number of misclassified objects varies in both experiments shows that the uniqueness of an object results in higher accuracy (see object 9 in both tables). Moreover, similar objects are often mistaken for each other (object 0 and 2). Additionally, an in-depth analysis of the results reveals that misclassifications occur when the model could not be trained on samples displaying a certain characteristic of an object (such as a handle of a mug) as these features naturally cannot be learned. This is the main point for the introduction of viewpoint variance as presented in our database.

VIII. Summary and outlook

In this contribution we present a publicly available benchmark database for RGB-D object instance recognition consisting of 60+ common household objects, which will be extended constantly by new objects, by our own initiative and upon request by other researchers. Each object is recorded with the Creative Gesture Camera resulting in RGB images of 640×480 resolution and ToF-based depth images of resolution 320×160. Each object is furthermore recorded with a step-wise variation of the viewpoint angles in azimuthal and polar coordinates of 36 degrees each, resulting in 100 snapshots per object. Invalid samples, e.g. due to artifacts or partial occlusion, are removed by a tree-cropping algorithm and complemented VOI cropping, yielding efficient object instance samples. With the construction described in this contribution we aim at establishing the single largest RGB-D database allowing extensive and precise algorithm development and testing, specifically targeted at indoor applications with a special focus on robotic scenarios. The experiments conducted for this contribution hint at the very obvious, but frequently omitted fact that viewpoint variance is indeed an important factor to be taken into account for creating object recognition models. The main shortcoming of related RGB-D reference databases is, to a large extent, the fact that near-perfect scenarios, i.e., limited viewpoint variations, are assumed. However, in a real world the positioning and alignment of objects within the environment cannot be predicted and has therefore be taken into account in all variants, when designing algorithms for any such task aimed at object manipulation.

We believe that our database represents a novel contribution for tasks such as robotic perception, instance recognition or machine vision in general. It has the potential to function as a new reference database providing additional benchmarks in areas such as computer vision or object recognition/reconstruction specifically due to the fact that we cover almost all viewpoint of an object exhaustively. The high-quality data cropping to depth segmentation supplements this novel database.

All data sets along with a documentation will be made available under www.gepperth.net/alexander/rgbd.php. Additional software and benchmark results will also be made available as part of the future work following this contribution.

REFERENCES

[1] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[2] Ashutosh Singh, Jin Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 509–516. IEEE, 2014.

[3] Thomas Kopinski, Alexander Gepperth, and Uwe Handmann. Time-of-flight-based hand posture database for human-machine interaction. In *International Conference on Control, Automation, Robotics and Vision, 2016 IEEE*, page tbd. IEEE, 2016.

[4] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, page 0278364911401765, 2011.

[5] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. Springer, 2013.

[6] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012*, pages 746–760. Springer, 2012.

[7] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

[8] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[9] Louis-Charles Caron, David Filliat, and Alexander Gepperth. Neural network fusion of color, depth and location for object instance recognition on a mobile robot. In *Second Workshop on Assistive Computer Vision and Robotics (ACVR), in conjunction with European Conference on Computer Vision*, 2014.

[10] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3384–3391. IEEE, 2008.

[11] Steffen Nissen. Implementation of a fast artificial neural network library (FANN). *Report, Department of Computer Science University of Copenhagen (DIKU)*, 31, 2003.