ijrr

The International Journal of
Robotics Research
31(8) 927–934
© The Author(s) 2012
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364912445831
ijr.sagepub.com
SAGE

# The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics

Alexander Kasper[1], Zhixing Xue[2] and Rüdiger Dillmann[1]

**Abstract**

*For the execution of object recognition, localization and manipulation tasks, most algorithms use object models. Most models are derived from, or consist of two-dimensional (2D) images and/or three-dimensional (3D) geometric data. The system described in this article was constructed specifically for the generation of such model data. It allows 2D image and 3D geometric data of everyday objects be obtained semi-automatically. The calibration provided allows 2D data to be related to 3D data. Through the use of high-quality sensors, high-accuracy data is achieved. So far over 100 objects have been digitized using this system and the data has been successfully used in several international research projects. All of the models are freely available on the web via a front-end that allows preview and filtering of the data.*

**Keywords**

Domestic robots, field and service robotics, service robots, humanoid robots, human-centered and lifelike robotics, manipulation planning, manipulation, recognition, sensing and perception computer vision

## 1. Introduction

In the field of service robotics, the ability to recognize, locate and manipulate objects in the environment is crucial. Without the ability to interact with its environment, a service robot is utterly useless. Currently, the basis for this ability is object models that are either learned online or pre-computed offline and then adapted and re-used at runtime. Online learning of object models often does not give the quality needed, or cannot capture object properties that are relevant because the appropriate sensor is not available. This is why offline-generated object models are still needed, as they can provide all of the necessary information at the maximum quality and accuracy.

The object-modeling setup presented in this paper has been constructed with this background in mind to simplify the generation of high-quality and high-accuracy models of typical household objects for use in service robotics. The data generated can be used in real-world scenarios as well as to test and evaluate new algorithms in object recognition and localization, grasp planning and execution.

This article is structured as follows: after a brief look at related work, the hardware setup is introduced. Then, the process developed to calibrate camera images relative to the three-dimensional (3D) data is described, which is then followed by a description of the capturing process as well as the data provided. The text concludes with a short section on how the data is deployed and some final thoughts about the presented and future work.

## 2. Related work

The main contribution of this work is the semi-automatic generation of 3D data with aligned two-dimensional (2D) image data. To the best of our knowledge, this fusion of 2D and 3D data specifically aimed at object recognition, localization and manipulation tasks of a service robot is unique. There is a multitude of similar work focusing exclusively on either 2D or 3D data sets however.

The generation of 3D data for a given real-world object via some method of digitization is a service that is currently provided by many companies. Most of these services focus on reverse engineering or quality assurance. A comprehensive list of companies in this sector is given by Wohlers Associates (2010).

[1] Humanoids and Intelligence Systems Lab, Karlsruhe Institute of Technology, Germany
[2] Interactive Diagnosis and Service Systems, Forschungszentrum Informatik, Germany

**Corresponding author:**
Alexander Kasper, Humanoids and Intelligence Systems Lab, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany.
Email: alexander.kasper@kit.edu

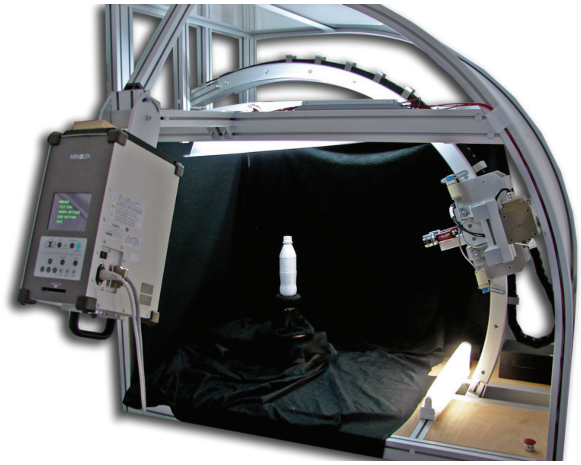

**Fig. 1.** The modeling center.

**Table 1.** Data sheet for the 3D digitizer: specifications for the Konica Minolta Vi-900.

| | |
|---|---|
| Measuring method | Triangulation light block method |
| AF | Image surface AF (contrast method), active AF |
| Image input range | 0.6–2.5 m (with different lenses) |
| Measurement input range | 0.6–1.2 m |
| Scan area | 111 × 84 mm to 710 × 533 mm, max. 1300 × 1100 mm |
| Sample time | 0.3–2.5 s |
| Number of output pixels | 640 × 480 (3D and color) |
| Geometrical resolution | $x = 0.17$ mm, $y = 0.17$ mm, $z = 0.047$ mm at 0.6 m |

When it comes to the combination of 2D and 3D data there are few systems able to provide this. The Stanford Spherical Gantry (Levoy 2010) is a system that was designed to automatically create images with specified lighting conditions. The construction allows for positioning of the cameras and lights around the object with a high degree of freedom. The setup does not explicitly include a device for 3D digitization, however.

The systems distributed by Cyberware Inc. (http://www.cyberware.com) are geared towards textured 3D object generation and are the closest to a combination of 3D and 2D data. The drawback here is that the systems do not produce intrinsically calibrated stereo images to be used by a recognition system.

Since we want to enable the fast and efficient modeling of a huge number of objects, our data set also needs to be compared to other large data sets. There exists a wide variety of 2D and 3D data sets that are provided as benchmarks. A well-known representative of a 3D data set is the Princeton Shape Benchmark (Shilane et al. 2004) which was created to compare and evaluate shape-matching algorithms. It contains a wide variety of different 3D models gathered from the web resulting in a very inhomogeneous data set consisting of models that vary greatly in accuracy, resolution and overall quality. A very new 3D data set is the RGB-D object dataset (Lai et al. 2011) which consists of 300 household objects digitized with a Kinect-style camera setup. The dataset is organized hierarchically according to WordNet (http://wordnet.princeton.edu) and contains video streams of three complete rotations at different viewpoints. This dataset is only available as a complete download and not searchable via a web interface. The quality of the 3D data is also not comparable to our sensor setup, since the Kinect-style camera's depth accuracy is an order of magnitude lower.

When it comes to 2D image data sets there is an even greater number of databases and benchmark test sets. Most of those sets, however, focus on varying the objects present in the images rather than the viewpoint around one specific object, like the PASCAL visual object classes data set (Everingham et al. 2010). One of the most comprehensive datasets for computer vision is ImageNet (Deng et al. 2009), which consists of several million images and is organized according to the WordNet hierarchy. Since these sets are focused on computer vision, mostly there is no accompanying 3D data available.

The grasp data that is bundled with each object is similar to the Columbia grasp database (Goldfeder et al. 2009) in that it contains joint angles for several hand poses for each object. It differs from the Columbia grasp database by providing tested grasps for specific, commercially available manipulators. This makes the data ready-to-use for all given objects and manipulators.

## 3. Hardware setup

The data to be collected for each object should consist of a triangulated 3D pointcloud as well as images from arbitrary viewpoints located on a sphere around the object. To achieve this a special rig was constructed that contains a 3D digitizer (Konica Minolta Vi-900), a turntable (Isel RF-1) and a pair of IEEE1394 cameras (Allied Vision Technologies Marlin 145C2) mounted on a sled that can move along a bent rail. Figure 1 illustrates this setup. In addition to the sensors and the turntable, three fluorescent lamps are mounted whose intensity can be controlled through an interface card in order to minimize specular or glossy highlights or to simulate various lighting conditions.

Table 1 summarizes the technical capabilities of the employed 3D digitizer. Being a commercial product, the Minolta Vi-900 performs with high accuracy and reliability. The turntable allows for semi-automatic generation of a scan series consisting of scans from different angles. With proper calibration on the turntable's turning axis, a pre-registration of the meshes can speed up the modeling process. The output of the scan process is a series of triangulated pointclouds of the parts of the object that are visible to the scanner in each successive scan. These are then post-processed using a custom application built on the

**Table 2.** Data sheet for the cameras: specifications for the AVT Marlin 145C2.

| | |
|---|---|
| Interface | IEEE1394a – 400 MBit/s |
| Resolution | 1392 × 1038 (YUV) |
| Sensor | SONY 1/2" progressive CCD |
| Frame rate | up to 10 fps at full resolution |

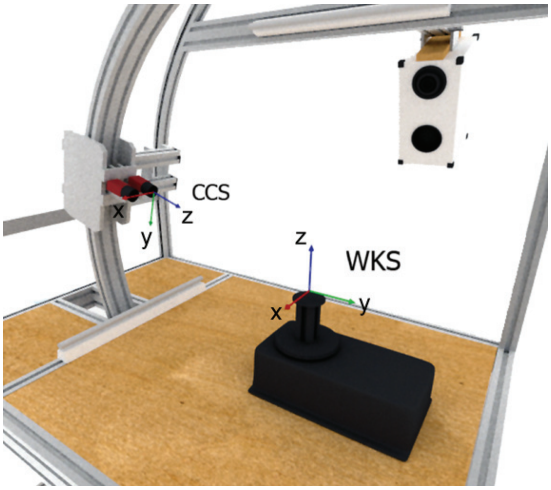

**Fig. 2.** World coordinate system (WCS) and camera coordinate system (CCS).

Rapidform DLL by Inus Technology (2010a). The software allows the registration and merging of the single scans, as well as a variety of state-of-the-art algorithms to improve mesh quality.

The stereo camera system consists of two IEEE1394 cameras with a resolution of 1392 × 1038 pixels. Table 2 lists the technical details of the specific camera model. The cameras are mounted on a sled that can move along a circular bent rail. The movement is controlled by a Schunk PowerCube. The rail is positioned so that the trajectory of the cameras rotates around an axis perpendicular to the rotation axis of the turntable. Together with the rotation of the turntable this allows the cameras to be positioned on a virtual hemisphere around the object. Through the calibration described in the next section these positions relative to the 3D data generated by the Minolta scanner are known.

Using the 3D data gathered, the information about the given object is enriched by pre-computed grasps for several commercially available manipulators, such as the Schunk Anthropomorphic Hand. This is done using custom grasp-simulation software.

## 4. Calibration

Even though the modeling center was manufactured with high precision, the exact positioning of the various hardware components cannot be derived from the schematics. In order to get the camera position relative to the 3D data of an object, several calibration steps have to be performed.

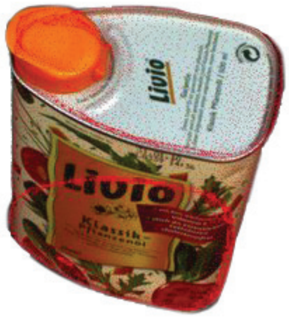

**Fig. 3.** Projection of a pointcloud into the object view for evaluation.

First of all the camera system needs to be calibrated intrinsically, and extrinsically to get the various parameters needed e.g. for stereo reconstruction. This is done using methods from the Integrating Vision Toolkit (IVT) (Azad 2010). The camera model and details of the calibration using a checkerboard pattern can be found in Azad et al. (2008).

With the intrinsic parameters, the transformation between image coordinates and camera coordinates is known. Next the position of the CCS in the WCS – which is defined by the Minolta digitizer – needs to be calculated. To do this an extrinsic calibration is carried out using the ARToolkit+ (Wagner and Schmalstieg 2007) by positioning a marker on the turntable. The ARToolkit+ is capable of calculating the camera position relative to this marker. Now the marker itself is scanned with the digitizer and its boundaries are marked in the pointcloud. Together this defines the transformation from WCS to the marker coordinate system and from there to the CCS. Images of the marker are taken from several positions along the camera rail and camera positions are calculated for each image. These points represent the trajectory of the camera rail, and together with the turntable rotation they can be used to calculate the camera positions for a given turntable and camera sled position. Figure 2 illustrates the positioning of the WCS and the CCS. To evaluate the results of this calibration process, two experiments were conducted. As a first test a sample object was digitized and the pointcloud was projected into the views to estimate how well they lined up with the image. The result of this can be seen in Figure 3. For the second test, a custom triangulation application was developed using IVT. Using scale invariant feature transform (SIFT) features as a basis for stereo triangulation, the application calculated a pointcloud of an object using several of the taken images and the intrinsic calibration. This pointcloud was then transformed into the WCS using the extrinsic calibration and was then compared to the mesh generated with the Minolta digitizer. This comparison was done using Rapidform Explorer (Inus Technology 2010b), which calculated the shortest distance of each point to the surface of the mesh. Histogram representations of these distances can be seen in Figures 4 and 5. Results show that the vast majority of the points are in

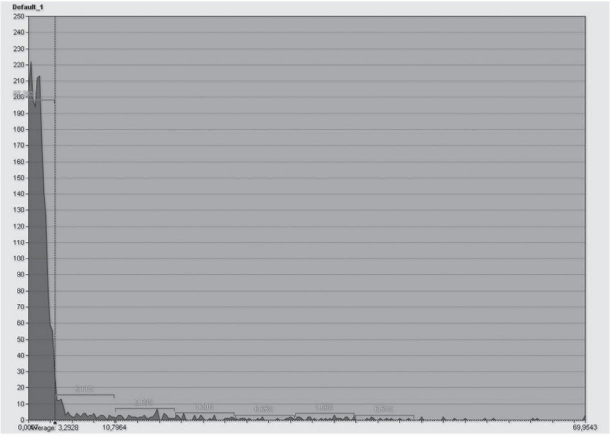

**Fig. 4.** Histogram of point distances from a triangulated point-cloud to a scanned mesh for the object 'yellow salt cube' (the horizontal axis is the distance from point to mesh, the vertical axis is the number of points for a given distance in mm).

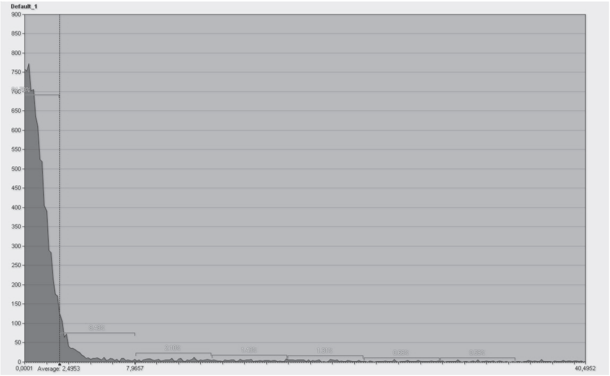

**Fig. 5.** Histogram of point distances from a triangulated point-cloud to a scanned mesh for the object 'danish ham' (the horizontal axis is the distance from point to mesh, the vertical axis is the number of points for a given distance in mm).

close proximity (fewer than 3 mm) to the scanned mesh, thus proving that the desired accuracy is reached.

## 5. Data capture

The actual data acquisition and processing is conducted in several consecutive steps. The complete workflow is presented in Figure 6.

### 5.1. 3D Data acquisition

The process starts with the acquisition of the 3D data using the scanner. Depending on the shape of the object, between 6 to 20 scans with different object positions are made, Figure 7 shows a sample result of this process. Since these pre-aligned single scans do not form a uniform mesh and contain holes and noise, the raw data needs to be further processed. The most commonly required operations are: outlier removal, filtering of spikes, fine-registration, merging, filling of holes and smoothing. The following paragraphs elaborate on these routines.

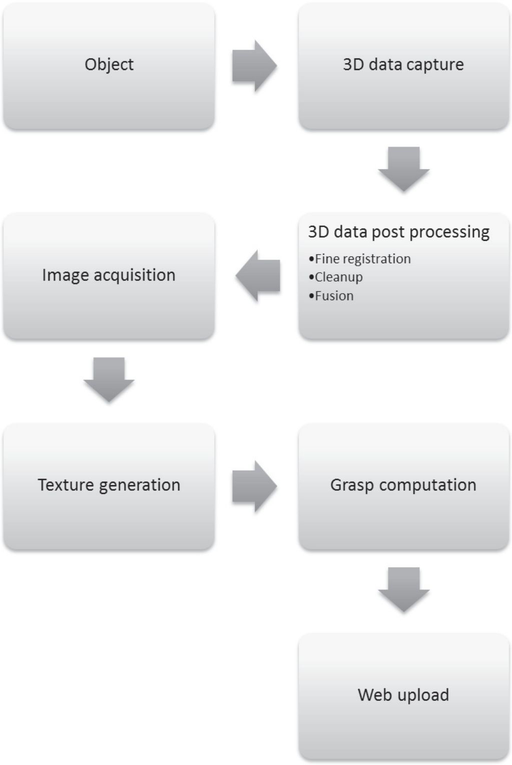

**Fig. 6.** Overview of the data capture workflow.

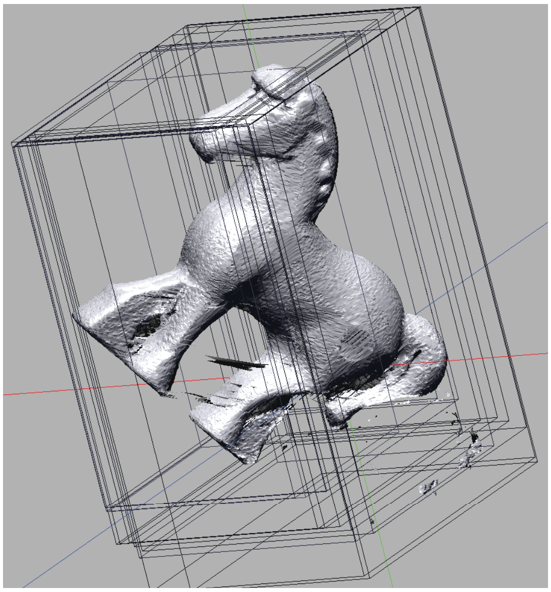

**Fig. 7.** Result of the raw 3D data capture: pre-registered single scans.

**Outlier removal** Outliers in the scanned data can have various sources. Reflective parts of the object, sharp corners or artefacts of the supporting structure are possible

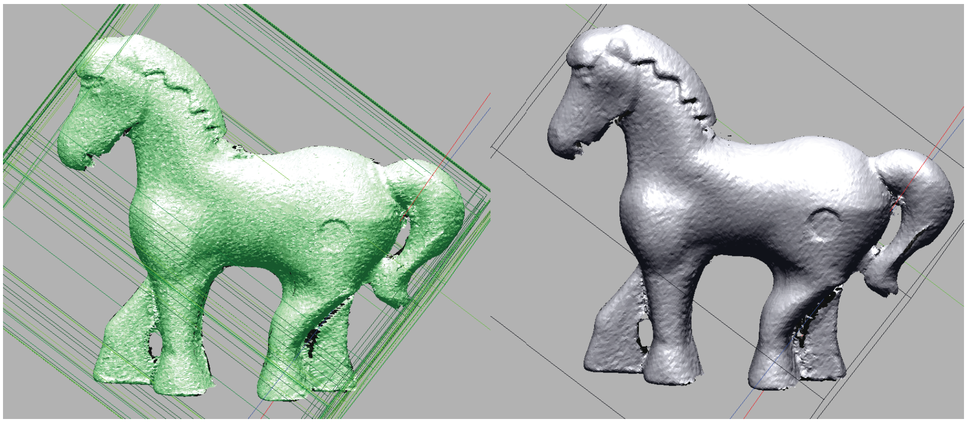

**Fig. 8.** Left: Result of fine-registration and the initial cleanup. Right: Result of the volume merge.

sources. Up to a certain cluster size these can be removed automatically, but manual correction is also often required.

**Spikes** Object surfaces located almost parallel to the view direction of the scanner produce spiked triangles that do not approximate the underlying surface very well. Additionally, sharp edges can produce fake geometry that mostly manifests in spikes. Before merging, these need to be removed. Fortunately spiked triangles are easily identified and thus can be removed automatically.

**Fine-registration** After the initial cleanup of the single scans, a fine-registration using ICP[1] can be performed which aligns the scans in the best possible way. If the input scans are pre-aligned well enough, this process can be conducted automatically.

**Merging** Well-aligned scans can be merged into a single mesh. This process removes duplicate triangles of the overlapping surfaces of the single scans to create one smooth surface completely automatically.

**Hole filling** Because of capture errors, incomplete surface coverage of the input scans or other processing artifacts, holes in the merged mesh may be present. Using surface interpolation these can be closed. During this process, manual input is required quite often, because the capabilities of the interpolation algorithms to cope with complex geometry are quite limited.

**Smoothing** To further reduce the noise introduced by the initial scans, the interpolation of neighbouring vertices and surface normals can be done to finalize the resulting mesh. This is a completely automatic operation.

Figure 8 shows two intermediate results of these operations. During this process, the need for further scans to complete coverage of the object may arise. These need to be registered to the existing scans and be post-processed as well. Finally, a last cleanup is performed as well as filling of holes in the volume merge. As a last step, the mesh is checked for non-manifold geometry which also needs to be cleaned (non-manifold geometry is often a problem for grasp-planning software operating on the mesh). The end result is a complete, 2D-manifold triangulated mesh, such as that depicted in Figure 9. The final post-processing is a reduction of the triangle resolution to create several versions of the mesh with different triangle counts, to suit applications that do not necessarily need high-resolution meshes (e.g. visualization). Generation of the 3D mesh by conducting the steps described above, usually takes between 30 minutes to 1 hour, depending on the complexity of the object. Convex objects are the most suitable for automatic digitization, while objects with a large number of cavities and inlets are the most difficult and require the most manual correction.

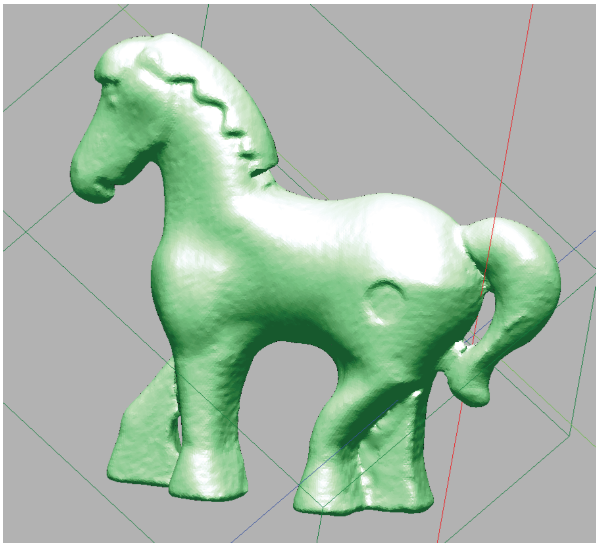

**Fig. 9.** Final result mesh of the 3D scanning stage.

## 5.2. *Image data acquisition*

When the mesh acquisition is finished, 2D images of the object are taken. To ensure matching between 3D and 2D data, a scan of the object is carried out and the previously generated mesh is fitted to this scan. Then automatic image generation is started. Default values here are every 10° for the turntable rotation as well as for the rotation of the sled. This results in 360 images for each camera representing the upper hemisphere around the object. If possible, the object is then turned upside down and the process is repeated to generate the lower hemisphere. The reference scan of the object in the upside-down position is registered to the original reference scan and the resulting transformation is then applied to the camera positions of the images from the lower hemisphere. This allows a complete capture of the object, but almost doubles the amount of data for each object. It is also not feasible for many objects as they cannot be placed in an upside-down position and thus has only been done for few objects.

The acquisition of the image data using the default values of the upper hemisphere takes approximately 20 minutes and is conducted completely automatically.

### 5.3. Texture generation

When all of the images have been taken, they are used to automatically generate texture information for the object. Inputs for the algorithm are: any number of images of the object with camera positions, intrinsic camera parameters and a triangulated mesh. For each input image, all triangles in the mesh are considered as possible projection targets. In a first step, a visibility check is performed to determine if the triangle in question is visible in the current image which can be done by comparing the normal of the triangle with the view direction associated with the image. As an optional step, a ray tracing based visibility check can also be incorporated here, to remove hidden triangles facing towards the camera (Kay and Kajiya 1986, Samet 1989). This eliminates a number of triangles for each image. The remaining triangles are then evaluated by calculating the angle between the normal and the view direction. If the angle is below a certain threshold, the triangle is associated with the corresponding image. Additionally, this angle criterion can be softened by neighbourhood relationships. This means, that the angle threshold is raised if a neighbouring triangle is already included in the projection. Since every triangle can only be projected into one image, the association of a triangle to a certain image can change during this process. This is done to maximize the size of continuous triangle areas projected into the same image, thus reducing the number of seams in the final texture.

Each triangle is then projected into its associated image and the normalized image coordinates of the vertices are used to create a new mesh definition including the texture mapping information. For each image the minimal region including all of its associated and projected triangles is determined. This region is then copied into a new image file that combines all views into a single texture file. Figure 10 shows four sample objects with finished textures, rendered using ray tracing software.

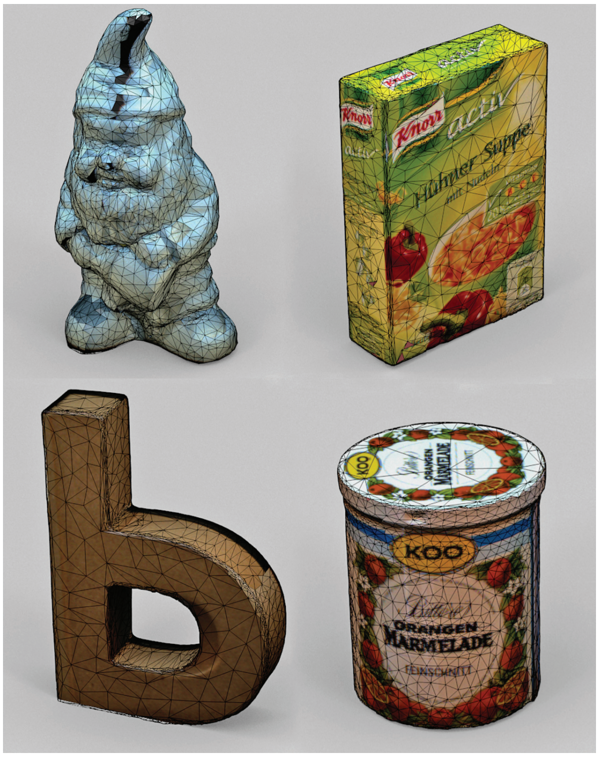

**Fig. 10.** Sample objects with textures and wireframes.

The duration of the texture-generation process is dependent on the mesh resolution. For all four default resolutions, approximately 5–10 minutes are currently necessary.

All together, complete processing of an object up to this stage takes 1–2 hours.

### 5.4. External object information and web upload

In a last step, the meshes are used to pre-calculate grasps for various robotic manipulators. The computation method to generate these grasps is described in Xue et al. (2009). Finally, all of the gathered object data is uploaded to the web database for dissemination.

## 6. Data description

During the various steps of data capture, several types of data are generated for each object:

- 3D Triangulated mesh
- 2D Image data
- Texture information
- Grasp information

The 3D data is available in different formats [Rapidform Model File (.mdl), Wavefront Object (.obj) and VRML97 (.wrl)] and resolutions (full ∼25,000, 5000 and 800 triangles). Each object has a unique name, which is used for all naming conventions in the data set. The 3D data files are named according to the following rules:

ObjectName Resolution[_tex].Format

where Resolution ∈ {Orig; 25k; 5k; 800} and Format ∈ {obj; wrl}. The image data is available in two different formats, Portable Network Graphics (PNG) and LZW-compressed Tagged Image File Format (TIFF). The naming of the files follows the convention:

ObjectName_isel_$\alpha$_amtec_$\beta$_LF_LB_LC_
[left|right].Format

where $\alpha$ is the angle of the turntable, $\beta$ denotes the angle of the camera sled, LF/LB/LC represent the light intensities for the front, back and center light and Format ∈ {png; tif}. There is also an accompanying XML file that connects the image files to the capture process. Explanation of the XML format can be found on the website http://his.anthropomatik.kit.ed/objectmodels/index.php?section=information#XML. Finally, the precomputed grasps are provided as XML data as well. At the moment, three popular robotic hands are supported: the three-fingered Schunk Dexterous Hand, the four-fingered Schunk Anthropomorphic Hand and the DLR/HIT Five-Finger Hand. A robot having one of these supported robotic hands may directly download the data and perform the grasp. Direct comparisons between different grasp-planning algorithms are also made possible by using the same object geometrical models.

The grasp data is documented in XML format. The ApproachPose node denotes the collision-free reachable hand pose before grasping, whereas the GraspPose node denotes the grasping pose of the robotic hand. Sixteen finger joints with unique names are listed for five fingers, each with three joints: base, proximal and distal. The thumb has an extra degree of freedom. The finger-joint positions before grasping, during grasping and after grasping are described in XML nodes: ApproachJoint, GraspPose and GraspOptimalJoint nodes, respectively. The GraspOptimalJoint node describes the optimal finger-joint positions that the fingers use to apply optimal grasping forces onto the object. Furthermore, each grasp is visualized in an image included in the database.

Since the dataset focuses on providing aligned 2D and 3D data of high quality, rather than facilitating symbol grounding or other semantic approaches for object recognition, the data is not yet organized hierarchically or into categories. We believe this to be future work when the size of the dataset has grown and additional different object categories are featured with a substantial number of representatives.

## 7. Data dissemination

In order to make the data available to a broad audience, a database with a web frontend has been established. It presents the data in a lucid way and allows for searches in the data set. In addition, a preview image is available and a news feed that notifies users of updates to the database. The website can be reached at http://his.anthropomatik.kit.edu/objectmodels.

## 8. Conclusion and future work

A system for the semi-automatic digitization of everyday objects has been presented. The system generates 3D and 2D data in a highly automized fashion, that allows the generation of models for a large number of objects in a reasonable amount of time. The generated data is highly accurate and suitable for applications in the fields of object recognition and localization as well as grasp planning, visualization and similar topics in the area of service robotics. Future work includes the addition of more objects to the data set. The modeling of other object properties is also something we would like to include. In particular, physical properties such as weight and material composition are of interest.

### Notes

[1] Iterative closest point.

### Funding

The generation of the object model data, the design and implementation of the underlying object representation and the interactive object modeling area were funded by the Deutsche Forschungsgemeinschaft in the course of the Collaborative Research Center 588 (SFB 588 *Humanoid Robots – Learning and Cooperating Multimodal Robots*), and partially by the German Bundesministerium für Bildung und Forschung in the course of the German Service Robotics Initiative *Desire*.

### References

Azad P (2010) Integrating Vision Toolkit. http://ivt.sourceforge.net.

Azad P, Gockel T and Dillmann R (2008) *Computer vision: Principles and practice*. Brentford: Elektor.

Deng J, Dong W, Socher R, Li LJ, Li K and Fei-Fei L (2009) ImageNet: A large-scale hierarchical image database. In: *IEEE conference on computer vision and pattern recognition (CVPR'09)*, Miami, USA, 20–25 June 2009, pp. 248–255. Piscataway: IEEE Computer Society.

Everingham M, Van Gool L, Williams CKI, Winn J and Zisserman A (2010) The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision* 88(2): 303–338.

Goldfeder C, Ciocarlie M, Dang H and Allen PK (2009) The Columbia grasp database. In: *IEEE international conference on robotics and automation (ICRA'09)*, Kobe, Japan, 12–17 May 2009, pp. 3343–3349. Piscataway: IEEE Computer Society.

Inus Technology (2010a) Rapidform DLL. http://www.rapidform.com/Contents/Product/Skin/ProductETC/category_id/53.

Inus Technology (2010b) Rapidform Explorer. http://www.rapidform.com/portal/default/Products/index? Category=Products_RapidformEXPLORER_Overview.

Kay TL and Kajiya JT (1986) Ray tracing complex scenes. In: *13th annual conference on computer graphics and interactive techniques (SIGGRAPH '86)*, Dallas, USA, 18–22 August 1986, pp. 269–278. New York: ACM Press.

Lai K, Bo L, Ren X and Fox D (2011) A large-scale hierarchical multi-view RGB-D object dataset. In: *international conference on robotics and automation (ICRA'11)*, Shanghai, China, 9–13 May 2011, pp. 1817–1824. Piscataway: IEEE Computer Society.

Levoy M (2010) Stanford spherical gantry. http://graphics. stanford.edu/projects/gantry.

Samet H (1989) Implementing ray tracing with octrees and neighbor finding. *Computers and Graphics* 13: 445–460.

Shilane P, Min P, Kazhdan M and Funkhouser T (2004) The Princeton shape benchmark. In: *shape modeling international (SMI'04)*, Genova, Italy, 7–9 June 2004, pp. 167–178. Washington DC: IEEE Computer Society.

Wagner D and Schmalstieg D (2007) ARToolKitPlus for pose tracking on mobile devices. In: *12th computer vision winter workshop (CVWW'07)*, St. Lambrecht, Austria, 6–8 February 2007, pp. 139–146.

Wohlers Associates (2010) 3D scanning & reverse engineering overview. http://www.wohlersassociates.com/scanning.html.

Xue Z, Kasper A, Zöllner JM and Dillmann R (2009) An automatic grasp planning system for service robots. In: *14th international conference on advanced robotics (ICAR'09)*, Munich, Germany, 22–26 June 2009, pp. 1–6. Piscataway: IEEE Computer Society.