A Robot Hardware

Our experimental setup consists of a robot arm, an end-effector mounted RGBD camera and a parallel jaw gripper. Our robot is a 7-DOF Kuka IIWA LBR. Mounted on the end-effector is a Schunk WSG 50 parallel jaw gripper. Additionally we mount a Primesense Carmine 1.09 RGBD sensor to the gripper body.

B Dataset Generation and Annotation

In order to reduce the human annotation time required for neural network training we use a data collection pipeline similar to that used in [4]. The main idea is to collect many RGBD images of a static scene and perform a dense 3D reconstruction. Then, similarly to [14], we can label the 3D reconstruction and propagate these labels back to the individual RGBD frames. This 3D to 2D labelling approach allows us to generate over 100,000 labelled images with only a few hours of human annotation time.

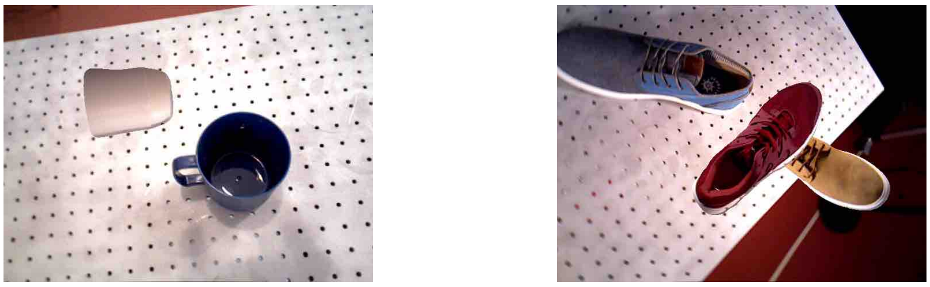
B.1 3D Reconstruction and Masking

Here we give a brief overview of the approach used to generate the 3D reconstruction, more details can be found in [4]. Our data is made up of 3D reconstructions of a static scene containing a single object of interest. Using our the wrist mounted camera on the robot, we move the robot's end-effector to capture a variety of RGBD images of the static scene. From the robot's forward kinematics, we know the camera pose corresponding to each image which allows us to use TSDF fusion [2] to obtain a dense 3D reconstruction. After discarding images that were taken from very similar poses, we are left with approximately 400 RGBD images per scene.

The next step is to detect which parts of the 3D reconstruction correspond to the object of interest. This is done using the change detection method described in [3]. In our particular setup all the reconstructions were of a tabletop scene in front of the robot. Since our reconstructions are globally aligned (due to the fact that we use the robot's forward kinematics to compute camera poses), we can simply crop the 3D reconstruction to the area above the table. At this point we have the portion of the 3D reconstruction that corresponds to the object of interest. This, together with the fact that we have camera poses, allows us to easily render binary masks (which segments the object from the background) for each RGBD image.

B.2 Instance Segmentation

The instance segmentation network requires training images with pixelwise semantic labels. Using the background subtraction technique detailed in Section B.1, we have pixelwise labels for all the images in our 3D reconstructions. However, these images contain only a single object, while we need the instance segmentation network to handle multiple instances at the test time. Thus, we augment the training data by creating multi-object composite images from our single object annotated images using a method similar to [23]. We crop the object from one image (using the binary mask described



(a) Mugs composite image **(b)** Shoes composite image

Fig. 8: Multi object composite images used in instance segmentation training

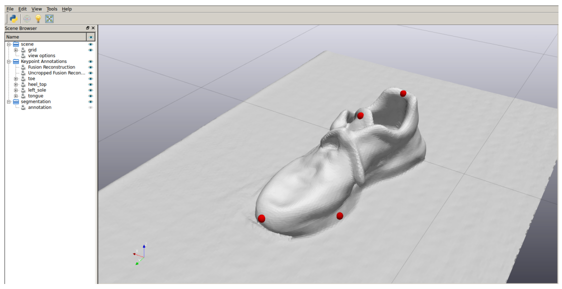


Fig. 9: A screenshot from our custom keypoint annotation tool.

in Section B.1) and paste this cropped section on top of an existing background. This process can be repeated to generate composite images with arbitrary numbers of object. Examples of such images are shown in Figure 8.

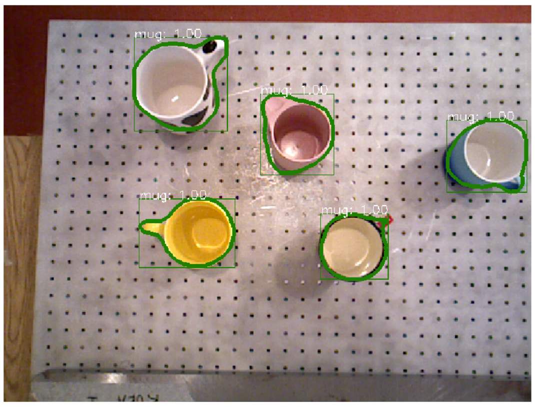
B.3 Keypoint Detection

The keypoint detection network requires training images annotated with pixel coordinates and depth for each keypoint. As mentioned in Section 3.2, we annotate 3D keypoints on the reconstructed mesh, transform the keypoints into the camera frame and project the keypoints into each image. To annotate the 3D keypoints on the reconstructed mesh, we developed a custom labelling tool based on the Director [15] user interface, shown in Figure 9. We labelled a total of 117 scenes, 43 of which were shoes and 74 of which were mugs. Annotating these scenes took only a few hours and resulted in over 100,000 labelled images for keypoint network training.

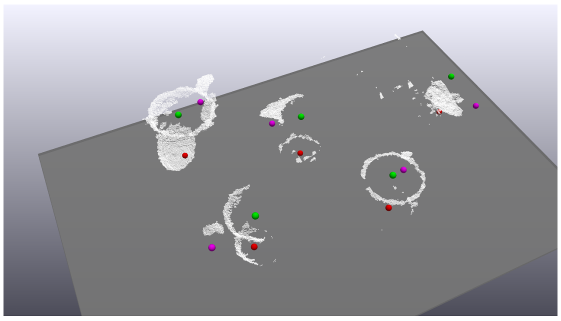
C Neural Network Architecture and Training

C.1 Instance Segmentation

For the instance segmentation, we used an open source Mask R-CNN implementation [16]. We used a R-101-FPN backbone that was pretrained on the COCO dataset [10].



(a) RGB image used for keypoint detections with Mask R-CNN annotations overlaid



(b) Keypoint detections

Fig. 10: 3D visualization of pointcloud and keypoint detections for the image from (a). The keypoints are colored as in Figure 6. The *top center* keypoint is green, the *bottom center* keypoint is red, and the *handle center* keypoint is purple.

We then fine-tuned on a dataset of 10,000 images generated using the procedure outlined in Section B.2. The network was trained for 40,000 iterations using the default training schedule of [16].

C.2 Keypoint Detection

We modify the integral network [25] for 3D keypoint detection. The network takes images cropped by the bounding box from MaskRCNN as the input. The network produces the probability distribution map $g_i(u,v)$ that represents how likely keypoint i is to occur at pixel (u,v), with $\sum_{u,v} g_i(u,v) = 1$. We then compute the expected values of these spatial distributions to recover a pixel coordinate of the keypoint i:

$$[u_i, v_i]^T = \sum_{u,v} [u \cdot g_i(u,v), v \cdot g_i(u,v)]^T \tag{8}$$

For the z coordinates (depth) of the keypoint, we also predict a depth value at every pixel denoted as $d_i(u,v)$. The depth of the keypoint i can be computed as

$$z_i = \sum_{u,v} d_i(u,v) \cdot g_i(u,v) \tag{9}$$

Given the training images with annotated pixel coordinate and depth for each keypoint, we use the integral loss and heatmap regression loss (see Section 2 of [25] for details) to train the network. We use a network with a 34 layers Resnet as the backbone. The network is trained on a dataset generated using the procedure described in Section B.3.

D Experiments

Figures 11, 12, 13 illustrate the results of experiments. These figures containing tiled images showing thee initial RGB image used for keypoint detection, along with an image of the object after running the kPAM pipeline. In the following sections we discuss more details related to the mug on shelf and mug on rack experiments.

D.1 Mugs Upright on Shelf

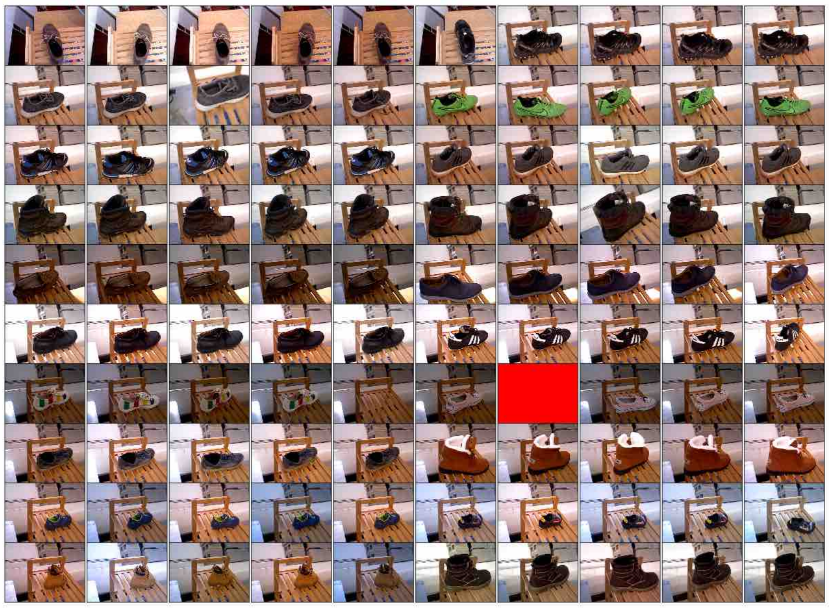
Results for the mug on shelf experiment are detailed in Figure 7. A trial was classified as a sucess if the mug ended up upright on the shelf with it's bottom center keypoint within 5cm of the target location. Out of 118 trials we experienced 2 failures. One failure was due to a combination of inaccurate keypoint detections together with the mug being torqued as it was grasped. Since we only have a wrist mounted camera we cannot re-perceive the object to compensate for the fact that the object moves during the grasping process. As discussed in Section 6 this could be alleviated by adding an externally mounted camera.

The other failure was resulted from the mug being placed upside down. Figure 14 shows the RGB image used for keypoint detection, along with the final position of the mug. As discussed in Section 5.2 this failure occurred because the keypoint detection confused the top and bottom of the mug. Given that the image was taken from a side view where the handle is occluded and it is difficult to distinguish top from bottom is understandable that the keypoint detection failed in this case. There are several ways to deal with this type of issue in the future. One approach would be to additionally predict a confidence value for each keypoint detection. This would allow us to detect that we were uncertain about the keypoint detections in Figure 14 (a). We could then move the robot and collect another image that would allow us to unambiguously detect the keypoints.

D.2 Hang mug on rack by its handle

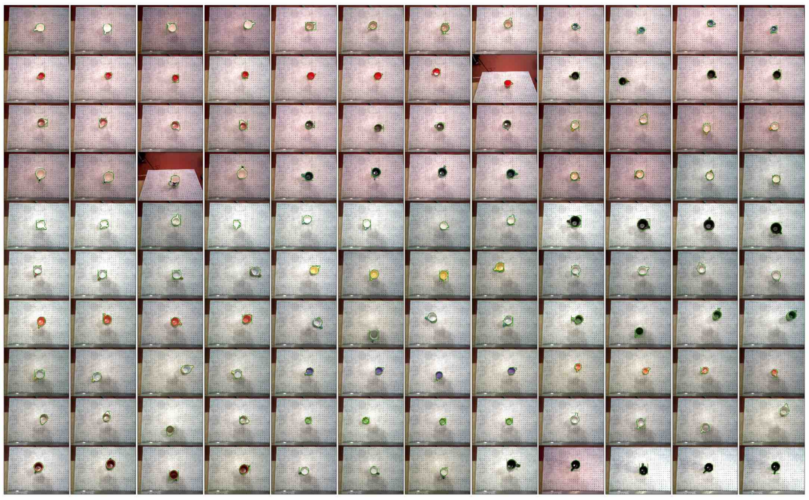
As discussed in Section 5.3 mugs were divided into two groups, *regular* and *small*, based on their size. A mug was characterized as *small* if the handle had a minimum

(a) RGB images used for keypoint detection in shoe on rack experiments

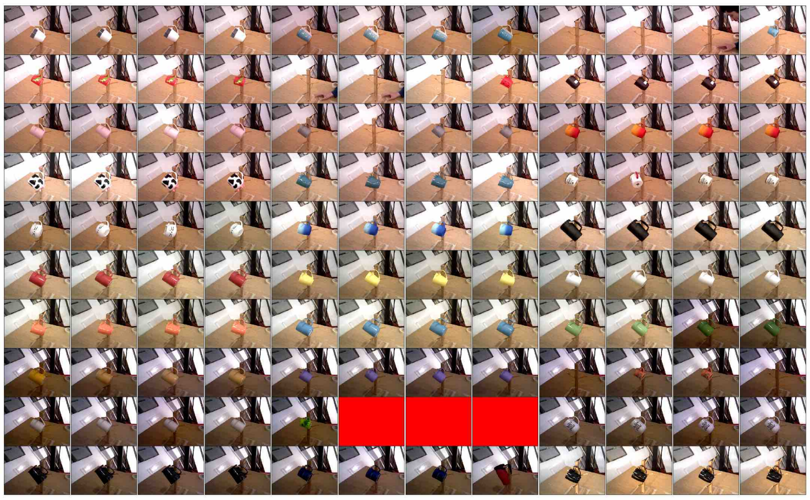


(b) Image of the shoe rack after running the kPAM pipeline. Red images indicate trials where no image of the final placement was captured due to an upstream failure of the pipeline causing the trial to be aborted.

Fig. 11: Before and after images of the shoe on rack experiment for all 100 trials.



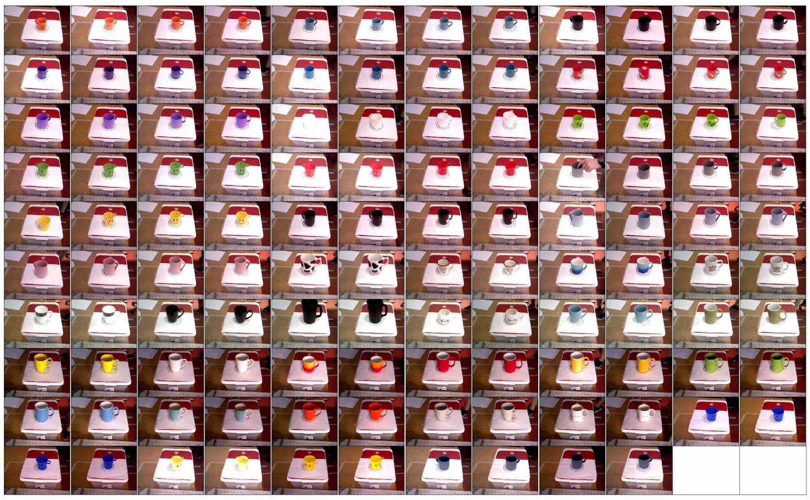
(a) RGB images used for keypoint detection in mug on rack experiments



(b) Image of the mug rack after running the kPAM pipeline. Red images indicate trials where no image of the final placement was captured due to an upstream failure of the pipeline causing the trial to be aborted.

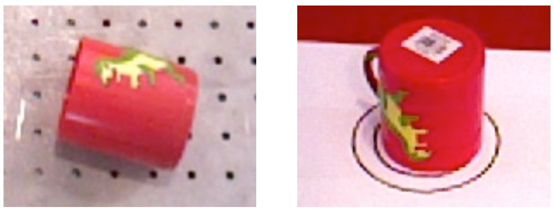
Fig. 12: Before and after images of the mug on rack experiments for all 120 trials.

(a) RGB images used for keypoint detection in mug on shelf experiments



(b) Image of the mug shelf after running the kPAM pipeline. Red images indicate trials where no image of the final placement was captured due to an upstream failure of the pipeline causing the trial to be aborted.

Fig. 13: Before and after images of the mug on shelf experiments for all 118 trials.



(a) (b)

Fig. 14: (a) The RGB image for the single failure trial of the mug on shelf task that led to the mug being put in an incorrect orientation. In this case the keypoint detection confused the top and bottom of the mug and it was placed upside down. (b) The resulting upside down placement of the mug.

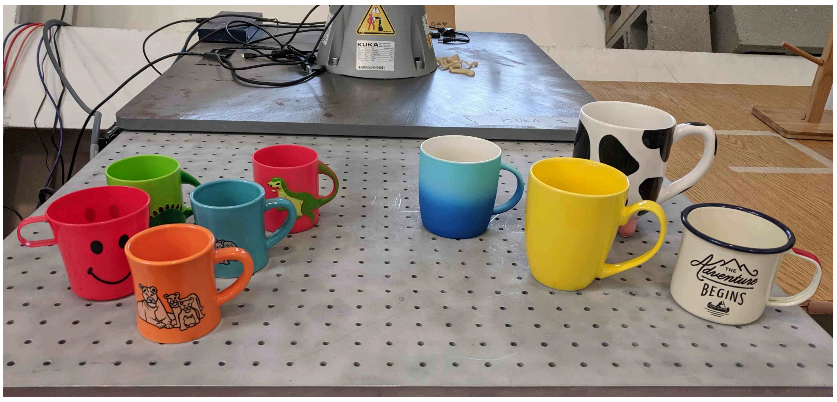


Fig. 15: The 5 mugs on the left are the test mugs used in experiment that were characterized as *small*. For comparison the four mugs on the right are part of the *regular* category.

dimension (either height or width) of less than 2cm. Examples of mugs from each category are shown in Figure 15. Mugs with such small handle sizes presented a challenge for our manipulation pipeline since hanging them on the rack requires increased precision.