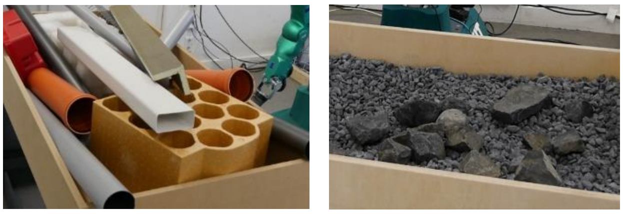
Grasp and Segmentation Strategies for Object Manipulation in Unknown Scenes based on Visual Features

To successfully interact with real world scenarios, robots must be able to cope with a large number of unstructured scenes that contain unknown objects and are cluttered. One of the most tasks in such scenes is picking up and placing objects, which requires the generation and execution of grasp candidates, which in turn often requires a segmentation of the scene. Even such a basic task is a challenge if the same system is to execute it successfully in many different scenes. The main reason for this is the high variance between scenes. Therefore, approaches relying on assumptions only work well in the specific class of scenes where this assumption is true and fail if this assumption is not true. A common assumption is that certain features of the scene, such as colors, are useful to derive some information about the scene, e.g. foreground/background segmentation. Other methods try to work with as few and general features as possible (e.g. shape of the scene's point cloud) and are therefore able to handle a greater variety of scenes. However, the results of these methods for a concrete scene are often worse than the results of a method that uses all available features.

A possible approach to solve this dilemma is to combine different methods depending on the current scene, to select and parameterize the currently used method. This project will focus on methods that are necessary for the above-mentioned pick and place task, such as the generation and execution of grasp hypotheses, segmentation and scene rearrangement actions to facilitate grasping.



In the first scene, color is a useful hint for segmentation, while it is useless in the second scene. The approach to generate and execute grasping actions differs as well.

Relevant research questions include:

- How can the robot decide on a method and parametrization to employ for the successful execution of a pick and place task if given a scene?
- How can an incorrect decision, which leads to the failure of task execution, be detected?
- How can the robot correct such an incorrect selection?

In this project, you will work with the humanoid robot ARMAR-6 as well as several robotics and machine learning tools:

- ArmarX (C++, Python): armarx.humanoids.kit.edu
- Tensorflow / PyTorch (Python): tensorflow.org / pytorch.org

Contact: Raphael Grimm (raphael.grimm@kit.edu)