# Detection of Textureless 3-D Objects

AUTHOR: ING. PETRA SVOBODOVÁ, SUPERVISOR: ING. JAN GAURA, PH.D.

*Technical University of Ostrava*

## Introduction

Topic of my diploma thesis is **detection** and **pose estimation** of textureless 3D objects.

The biggest motivation is fact, that many similar algorithms are based on comparing textures. But we have many objects without textures in real world [1]. Detection of textureless objects can be used in **industry** or **robotics**. There is often a need to manipulation with tools or products which are without textures.

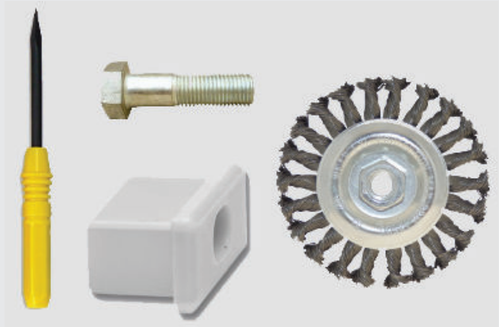

Figure 1: Example of textuleress objects which you can meet in industry

Another difference against similar algorithms is that objects are captured in **different sizes** and from **different angles of view**. They usually need many training images per one object position. It takes a lot of time. Our approach removes large number of training pictures. So we need only **one training image per one object position**.

Goals of my work are **implementation algorithm** for fast detection of textureless 3D objects and **localization** of the object including value of the similarity.

## Approach

In the picture below [2] there is a **training dataset**. As you can see, it represents a large number of training pictures. Actually, we need only one image per one object position. This is the main difference against other similar algorithms.
All objects in training dataset are **without texures**. It means that only information about objects are their **edges**.

Algorithm finds to the object the most similar pattern from traing dataset.
It also **draws a contour** of the most similar pattern and **compute the similarity** between object and pattern from training dataset.

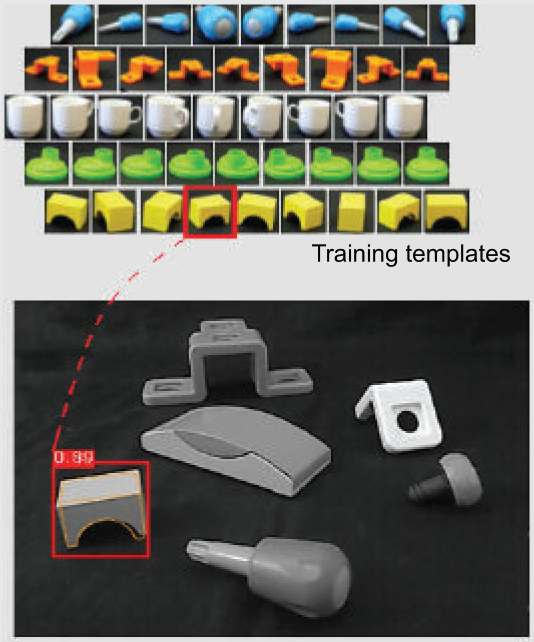


Figure 2: How algorithm works

## Methods

Our approach is **edge-based**. So first step is **detect edges** and define reference points in regular grid [3]. Then we compute two **features - distance and orientation** to describe image.

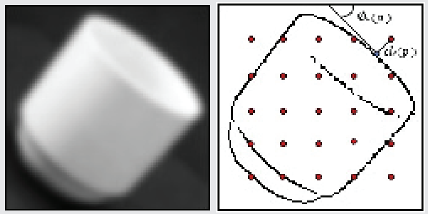

Figure 3: Original image (left) and edge image with reference points (right)

Algorithm quantizes features of all training templates and puts them to index table [4]. Searching similar features by index table is fast. So with this step we remove more then 90% of background.

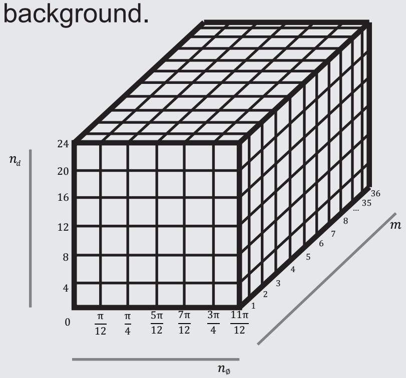

Figure 4: Index table for matching features ($n_d$ - quantified value for distance, $n_o$ - orientation, $m$ - number of reference points)

Second step is more time-consuming. We compute oriented chamfer matching to get rid of some less similar scanning windows. After this stage we have only few candi-dates for searched objects.

Algorithm has some improvements. One of them is **selecting discriminative edges** according to stability of viewpoint and stability of orientations.

We remove 40% of pixel edges from long lines and expect better results.

At the end, we receive few candidates. We filter this set of potentially searched objects by method called non-maxima suppression. The result with highest OCM score is our best match.

## Results

Algorithm runs in four settings:
- original edges & improvement OCM,
- discriminative edges & improvement OCM,
- original edges & original OCM,
- discriminative edges & original OCM.

We tested algorithm in the CMP-8objs dataset.

The average precision of detector with all improvements is **78,25%**. It is higher then without them. It could seems low, but there are objects with precision more then 90%. But on the other hand, there are objects with difficult edges. Their precision of detection is lower then 50%. It means that value of accuracy depends on the type of object.

Average recall for all objects and for both improvements is 68,75%. It means that algorithm is working and both improvements have better results, but there is still space for improvement.