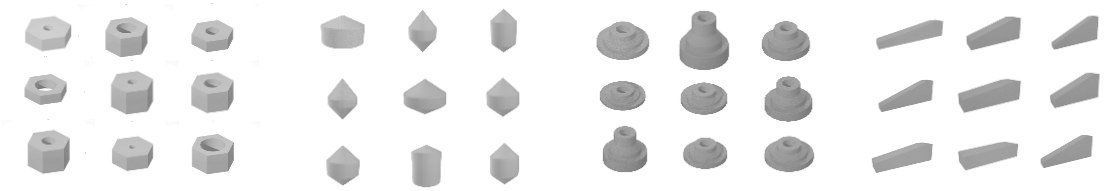
Parametric Dataset for 6DoF Pose Estimation

Introduction

Our dataset consists of 4 shape templates TN06, TN16, TN34 and TN42, selected from Zeng's parametric database. The dataset has 386880 scenes depicting various numbers of objects in stacked scenarios, which is designed for training and evaluation for 6DoF pose estimation in parametric shapes stacked scenarios.



(a) TN06 (b) TN16 (c) TN34 (d) TN42

Fig. 1. Four shape templates

Construction process

Physics simulation

The part instances can be generated from the shape templates. For each part instance, first, we simulate the free-falling process of 1 object, record the scene and clear the object in the scene. Then, we simulate the free-falling process of 2 objects, record the scene and clear the objects in the scene. We can get one cycle through repeating the steps above for each part instance until N (drop limit) objects in a scene is recorded. To increase the richness of the dataset, multiple cycles can be generated repeatedly as required. In our dataset, we set N to 60.

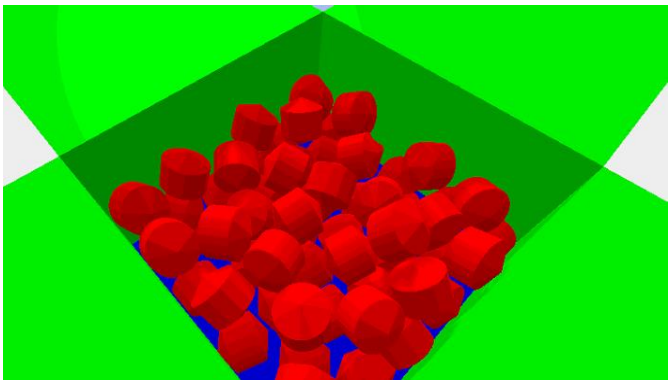


Fig. 2. Physics simulation of TN16 stacked scene

Render

After rendering physics simulation result, we will get depth images and segmentation images. The depth images contain depth in the scene and are saved as png format. The segmentation images are saved as exr format. We use Blender to generate a colored segmentation image and the pixels belong to the same object have the same color.

Ground truth

A ground truth file corresponds to a scene, annotating the labels of each object in the scene, such as class name, segmentation ID, translation, rotation matrix, visibility and parameters.

Dataset description

Structure

training_set: The training set includes 29 cycles of 4 shape templates (TN06, TN16, TN34, TN42 are sampled 64, 64, 16, 64 part instances respectively) with a total of 361920 scenes depicting various numbers of objects in stacked scenarios.

learning_test_set (i.e. L-dataset): To test the learning ability, the learning test set includes 1 cycle of 4 shape templates (TN06, TN16, TN34, TN42 are sampled 64, 64, 16, 64 part instances respectively) with a total of 12480 scenes is constructed, whose instances' parameters are the same as training set.

generalization_test_set (i.e. G-dataset): To test the generalization ability, the generalization test set includes 1 cycle of 4 shape templates (TN06, TN16, TN34, TN42 are sampled 64, 64, 16, 64 part instances respectively) with a total of 12480 scenes is constructed, whose instances' parameters are different from training set, i.e. about 2%-37% difference in each parameter value.

Archive description

model: This folder contains the 4 shape templates, and the part instances' models are obtained by sampling on manifold surface and saved as obj format. We represent the part instances as the serial numbers. Taking TN16 as an example, the 64 part instances we sampled are respectively represented as 0, 1, 2, ..., 63.

depth_images: Depth information of the objects for each scene.

segmentation_images: Segmentation of the objects for each scene.

gt: Ground truth annotations for each scene.

parameter.json: Camera parameters are described within this file which provides camera parameters for the parsing of the depth images and the segmentation images, and turning them into point cloud.

File naming description

The name of the scene data files in each cycle follows the format "part instance name_*N* (drop limit)". For example, the file name "5_002" means the scene contains 2 objects of the part instance 5.

Converting to point cloud

The depth images store the linear depth where each pixel value is $depth[u, v]$. The maximum pixel value max_val_in_depth corresponds to the maximum interception depth. Therefore, the linear depth is calculated as follows:

$$Z_{line} = clip\_start + \frac{depth[u, v]}{\text{max\_val\_in\_depth}} \times (clip\_end - clip\_start)$$

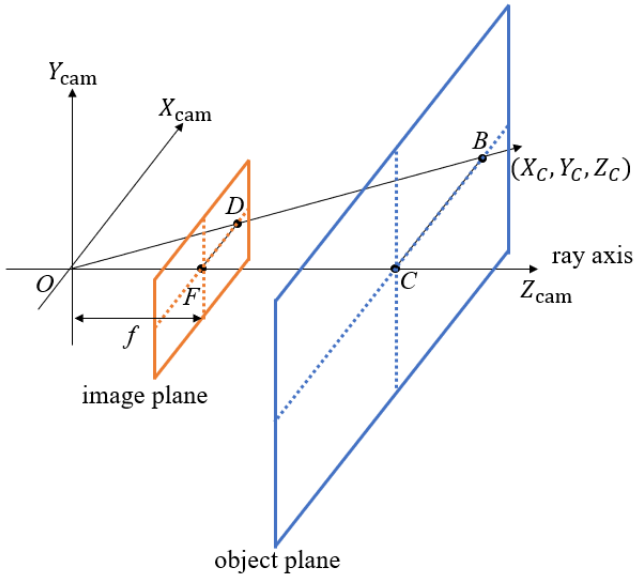
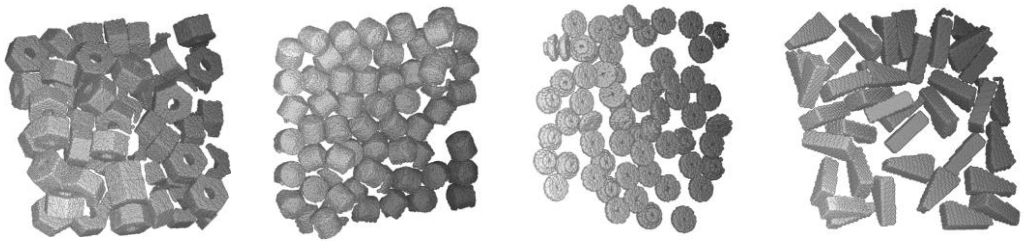


Fig. 3. Visual coordinate system model

The linear depth (*OB*) needs to be converted into the vertical depth (*OC*), which can be calculated by similar triangles. To get the point cloud, (X_C, Y_C, Z_C) in camera coordinate system is computed from the value of the depth image at a pixel (u, v) as follows:

$$\begin{cases} Z_C = \dfrac{Z_{line}}{\sqrt{1 + \left(\dfrac{u - c_u}{f_u}\right)^2 + \left(\dfrac{v - c_v}{f_v}\right)^2}} \\ X_C = \dfrac{Z_C}{f_u}(u - c_u) \\ Y_C = \dfrac{Z_C}{f_v}(v - c_v) \end{cases}$$

where the camera coordinate system is in unit m and the pixel coordinate system is in unit pixel.



(a) TN06 (b) TN16 (c) TN34 (d) TN42

Fig. 4. Point cloud of the stacked scenario

Download

The dataset can be downloaded archives:

https://pan.baidu.com/s/1Eyg9Vpa4Wa2mGfP_cPm9_A. (Password: hc1q)