Bridging Realities: Training visuo-haptic object recognition models for robots using 3D virtual simulations

Supplemental Materials

A. Virtual Haptic Capture System – Ray Generation

The main goal of the section is to generate the line segments, $\overline{F_l T_l}$, $\overline{F_m T_m}$ and $\overline{F_r T_r}$, that correspond to the paths of the fingers of the robot to touch the object (see Fig. 20). Where, F_l, F_m, and F_r , are the initial locations/points of the end-effectors of the fingers. And where, T_l, T_m, and T_r , are the target points of the fingers. First, we generate the line segment, $\overline{F_m T_m}$. After which, we can calculate the two other parallel lines, $\overline{F_l T_l}$, and $\overline{F_r T_r}$.

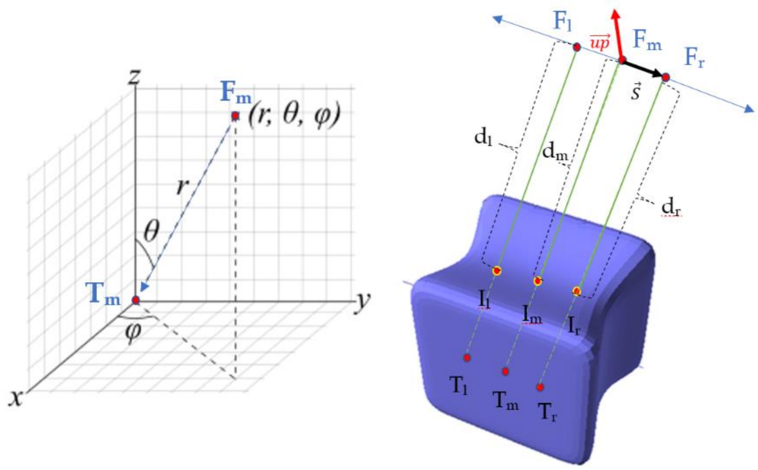


Fig A.1 The main 3 rays that represent the path of the fingers of the robot ($\overline{F_l T_l}$, $\overline{F_m T_m}$ and $\overline{F_r T_r}$); and the representation in the spherical coordinate system with the target point (T_m) as the origin

Once we have these three line segments we can compute the intersection or contact points, I_l, I_m, and I_r. This allows us to calculate the distance between the initial points of the fingers to the contact points, which will ultimately allow us to calculate the contact time of the fingers based on the velocity ($P_{velocity}$).

We set the target point of the middle finger, T_m, to the center of the bounding box of the stimuli (see Fig. A.1) To obtain the point, F_m, we generate two random angles, θ and φ (see Fig. A.1). The parameter, P_{step_angle}, determines the precision of the random angles that will be generated. For example, if P_{step_angle} is 45° then the possible random angles for θ and φ would only be the values 0°, 45°, 90°, etc. Using the spherical coordinate system and the angles of θ and φ, and setting *r* be to the length of the diagonal of the bounding box, we can determine a point F_m (see Fig. A.1). The point, F_m corresponds to the starting point of the middle finger of the robot.

To be able to generate the points F_l and F_r from the center finger point F_m, we need to calculate a side vector, $\vec{S}$ (see Fig. A.1). The vector $\vec{S}$ is computed as the cross-product of $\overrightarrow{F_m T_m} \times \overrightarrow{up}$, this creates a vector orthogonal to both vectors. The vector, $\overrightarrow{up}$ is defined as the *z-axis (0,0,1)* except in the degenerate case that it is parallel with $\overrightarrow{F_m T_m}$. In this case, we use the *y-axis (0,1,0)*. The side vector is normalized to a unit vector and rotated around using $\overrightarrow{F_m T_m}$ as the axis of rotation by the randomly generated angle, α. This simulates the rotation of the robot's hand in the real world at its wrist. Therefore, F_l and F_r can be calculated as:

$$\mathrm{F_l} = \mathrm{F_m} - \mathrm{P_{f\_dist}} \cdot \vec{S} \text{ and } \mathrm{F_r} = \mathrm{F_m} + \mathrm{P_{f\_dist}} \cdot \vec{S}$$

similarly, we can compute the target points, T_l and T_r as:

$$\mathrm{T_l} = \mathrm{T_m} - \mathrm{P_{f\_dist}} \cdot \vec{S} \text{ and } \mathrm{T_r} = T_m + \mathrm{P_{f\_dist}} \cdot \vec{S}$$

We finally have the three line segments that correspond to the path of the 3 fingers of the robot, these are, $\overline{F_l T_l}$, $\overline{F_m T_m}$ and $\overline{F_r T_r}$.

B. Experiments using Everyday Objects

In this section, we use ten models from the Princeton Mesh Benchmark [Chen et al.(2009)]:

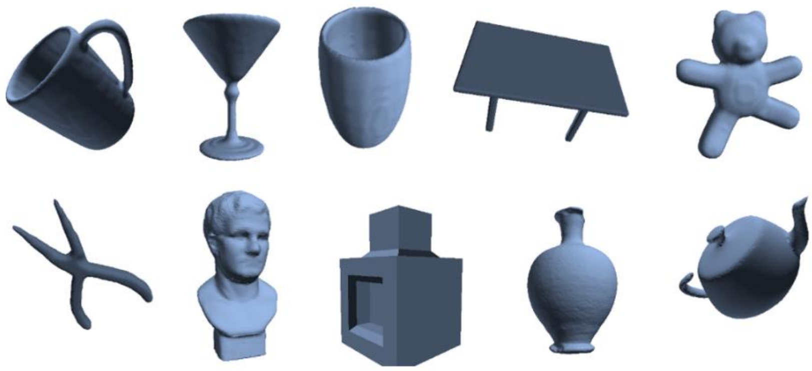
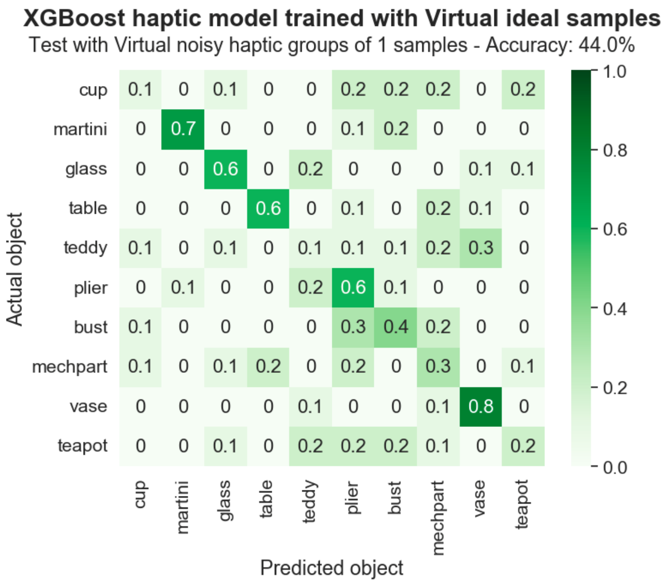
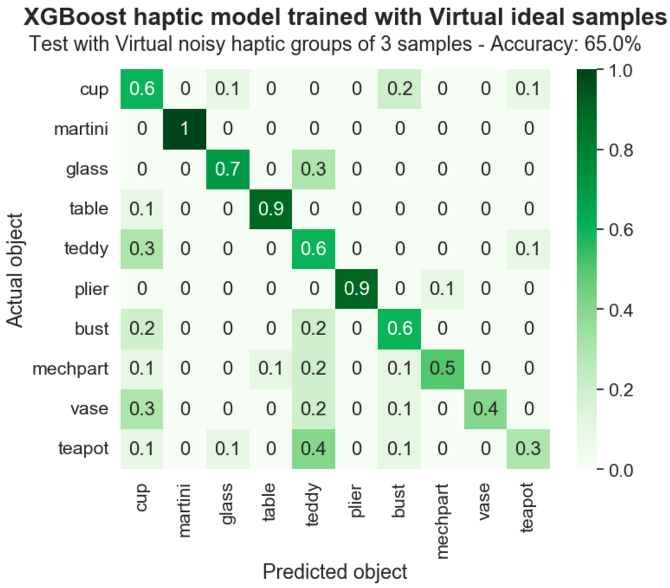
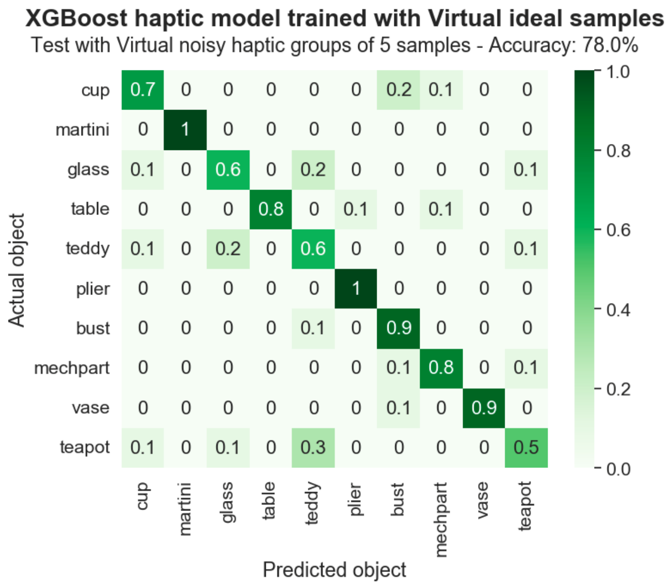


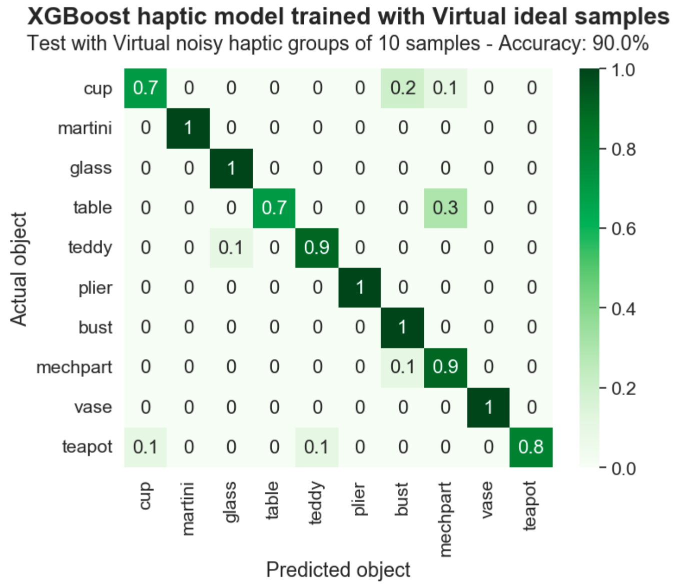
Fig. B.1 Selected models from the Princeton Mesh Benchmark [Chen et al.(2009)]: cup, martini, glass, table, teddy, pliers, bust, mechanical part, vase, teapot.

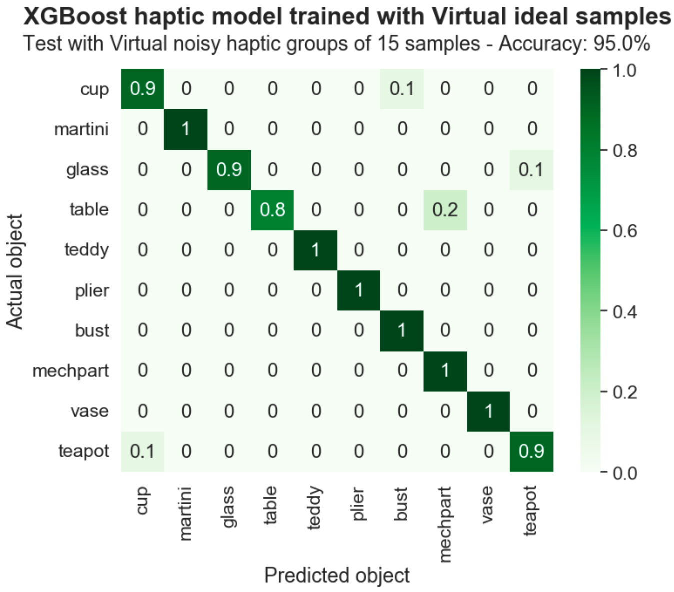
For this dataset, we carried out a **haptic classification** with different amounts of test samples. The models were trained with virtual ideal samples and tested with noisy virtual samples. As in our primary dataset, we introduced Gaussian noise of 0.04799 to the virtual data.



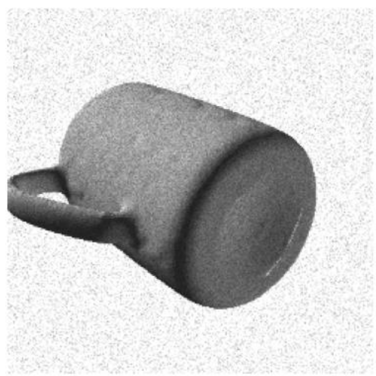




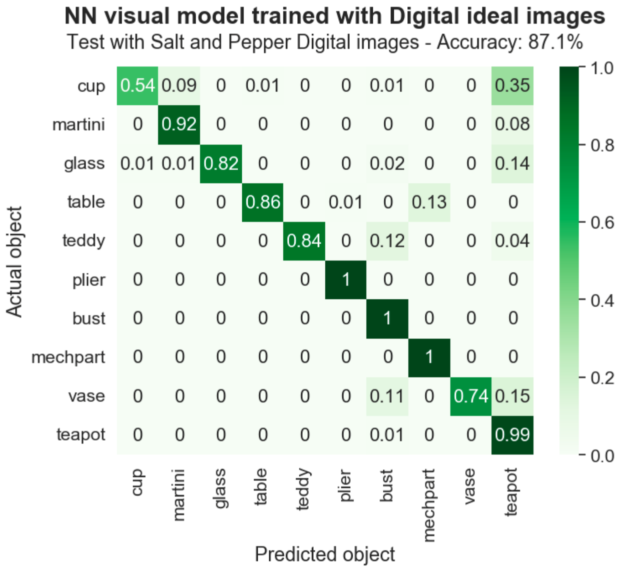


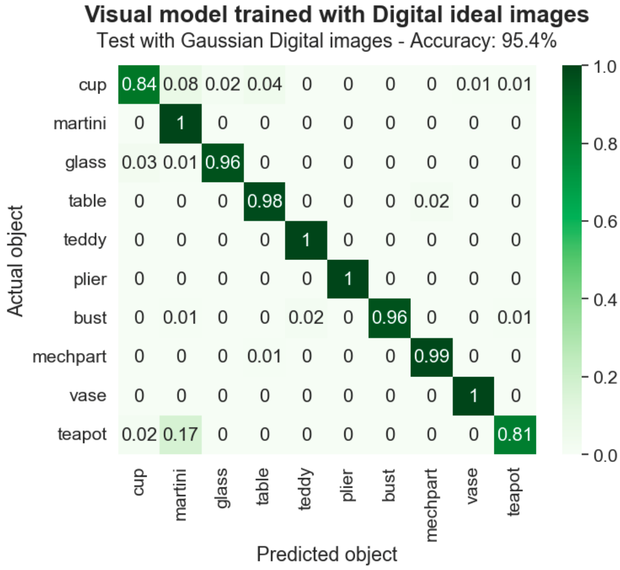


For this dataset, we also carried out a **visual classification.** Since we don't have any real data for this dataset, we introduced different types of noise to evaluate the trained models.

 Virtual ideal image	 Virtual image with salt and pepper noise	 Virtual image with gaussian noise

So as for the haptic classification, we trained a model with virtual ideal images and then tested the model with virtual images with the two types of noises.





Finally, we evaluated the **multimodality** of this dataset with Gaussian virtual images and noisy haptic virtual samples. As can be seen in the matrix below, multimodality showcases an accuracy higher than the one we observe with any of the modalities working independently.