OBJECTFOLDER 2.0: A Multisensory Object Dataset for Sim2Real Transfer (Supplementary Materials)

Ruohan Gao[1]* Zilin Si[2]* Yen-Yu Chang[1]* Samuel Clarke[1]
Jeannette Bohg[1] Li Fei-Fei[1] Wenzhen Yuan[2] Jiajun Wu[1]
[1]Stanford Univeristy [2]Carnegie Mellon University

The supplementary materials for [9] consist of:

A. Supplementary Video.

B. Dataset Details.

C. Details for Vision, Audio, and Touch Simulations.

D. Implementation Details for Implicit Neural Representation Networks.

E. More Qualitative Examples of Mulitisensory Data.

F. Experiment Set-up Details for the Three Sim2Real Tasks.

G. Additional Sim2Real Experiment on Grasp Stability Prediction.

H. Additional Analysis on Audio Simulation

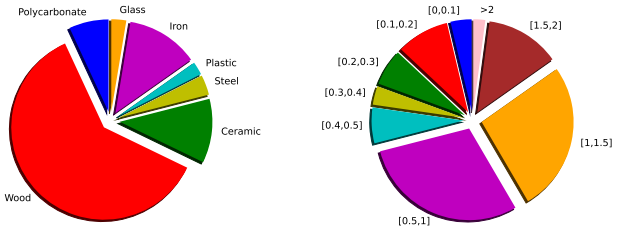
A. Supplementary Video

In the supplementary video, we show 1) the motivation and goal of OBJECTFOLDER 2.0; 2) the visualization of the 1,000 objects from our dataset; 3) examples of the multisensory data for some sample objects and comparisons with OBJECTFOLDER 1.0 [8]; 4) demos of real-world tactile-audio contact localization experiments; 5) demos of real-world visuo-tactile shape reconstruction experiments.

B. Dataset Details

The 1,000 objects in OBJECTFOLDER 2.0 are all of approximately homogeneous material property, and the material types include ceramic, glass, wood, plastic, iron, polycarbonate, and steel. See Fig. 1a for the material type distribution of the objects. The objects in our dataset are also of diverse scales (length of the longest side of the axis aligned bounding box enclosing the object). Fig. 1b shows the distribution of object scales in meter. Both the scale and material type of the objects are used in modal analysis for realistic audio simulation.

*indicates equal contribution.



(a) Material type (b) Object scale

Figure 1. Statistics of materials types and object scales for the 1,000 objects in OBJECTFOLDER 2.0.

C. Details for Vision, Audio, and Touch Simulations

Vision: We use Blender's Cycles path tracer [1] to render images. For each object, we first normalize it into a unit cube and use a point light source at a random location on a unit sphere with radiance of (1, 1, 1). We then render images of the object on a white background from camera viewpoints randomly sampled on a full sphere with a radius of 2.5. We render 500 images each for training, validation, and testing.

Audio: We first tetrahedralize each object's surface mesh using a technique shown to produce high-quality tetrahedralizations from meshes found in the wild [12]. See Fig. 2 for a comparison of the tetrahedron meshes that we use and the volumetric hexahedron meshes used in OBJECTFOLDER 1.0. We can see that the tetrahedral mesh captures finer features and surface curvature at the same representation size, so it can more accurately model the acoustic properties of the objects. We then use the material parameters as shown in Table 1 for each object and perform modal analysis of the object mesh with Abaqus FEA software [4], using second-order mesh elements for the analysis. Whereas OBJECTFOLDER 1.0 uses first-order elements for its modal analysis, the second-order elements we use in OBJECTFOLDER 2.0 capture the elastic deformations of the object at a higher-resolution, producing even more accurate

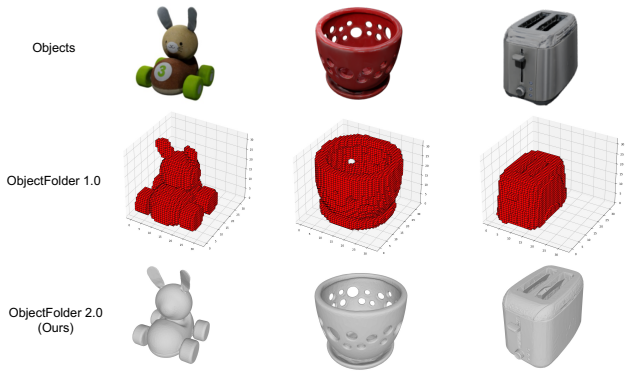


Figure 2. Comparison of the volumetric hexahedron meshes used in OBJECTFOLDER 1.0 and the tetrahedron meshes used in OBJECTFOLDER 2.0 for three representative objects.

analysis results [5]. From this modal analysis, we obtain frequencies of each mode, as well as the initial gains of each mode when contacted at each point on the surface mesh by a unit force in each axis direction. Using the frequencies of each mode along with material parameters, we estimate the dampings of each mode according to a Rayleigh damping model [19].

Touch: We use the geometric measurement from a GelSight tactile sensor as the touch reading. GelSight is a vision-based tactile sensor that interacts the object with an elastomer and measures the geometry of the contact surface with high spatial resolution images by capturing the deformation of the elastomer with an embedded camera. To simulate the tactile sensing, we need to simulate 1) the deformation of the contact, and 2) the optical response to the deformation. Because GelSight provides richer and denser contact signals compared to most tactile sensors, its simulation can be potentially used to synthesize the readings from other tactile sensors [14, 17].

We adopt a two-stage approach to render realistic tactile signals. Firstly, we simulate the sensor-object interaction to render the deformation map, which is constructed from the object's shape in the contact area and the gelpad's shape in the non-contact area to represent the local shape at the point of contact. Then we utilize the optical simulation of Taxim [20], an example-based tactile simulation model. Taxim simulates the optical response to the deformation of the soft elastomer with a polynomial lookup table. This table maps the deformed geometries to pixel intensity sampled by the embedded camera. Finally, the simulation model is calibrated with examples from a real-world GelSight tactile sensor to reduce the Sim2Real transfer gap.

D. Implementation Details for Implicit Neural Representation Networks

VisionNet: We follow the prior work on object-centric neural radiance field [10] to represent each object as a 7D object-centric neural scattering function (OSF). OSF uses a eight-layer MLP with 256 channels to predict the density value σ, which is view invariant; and a four-layer MLP with 128 channels to predict the fraction of the incoming light that is scatterd in the outgoing direction $\rho = (\rho_r, \rho_g, \rho_b)$. Following KiloNeRF [18], to accelerate neural rendering, instead of using a single MLP to represent the entire object, we represent each object with a large number of independent and small MLPs each responsible for a small portion of the object. We use a small MLP that has four layers with 64 channels to predict σ; and a one-layer MLP with 64 channels to predict ρ. We first train an ordinary OSF model and distill the knowledge of this teacher model into the KiloOSF student model. The KiloOSF model is trained such that its output matches the output of the teacher model. We optimize the student model's parameters by using an L_2 loss between the values predicted by the student model and those obtained from the teacher model. We also follow the sampling strategy in [18]. We set the network grid resolution to be $16 \times 16 \times 16$, and the threshold τ for occupancy grid extraction to be 10. We use the Adam optimizer with learning rate of $5 \times e^{-4}$, a batch size of 8192, and the same learning rate scheduler as [16]. See [18] for more details.

AudioNet: For each axis direction, we train a separate branch to encode the corresponding gains for all vertexes. Each branch is an eight-layer MLP with 256 channels. For each object, we normalize the values of gains to $[0, 1]$ as the prediction target. We use the same positional encoding scheme as in [16].

TouchNet: We simulate the sensor-object interaction with Pyrender [3] to render deformation maps using OpenGL [2] with GPU-acceleration. During training, for each surface location of the polygon mesh of the object, we ramdonly sample a rotation angle within $\pm 15°$ and a pressing depth (gel deformation) in the range of 0.5-2 mm, then we obtain a deformation map of dimension 120×160 that captures the local geometry information. TouchNet is also an eight-layer MLP with 256 channels that takes the spatial coordinate (x, y, z) of the vertex, a 3D unit contact orientation parametrized as (θ_T, ϕ_T), gel penetration depth p, and the spatial location (w, h) in the deformation map as input. The output is the per-pixel value of the deformation map for the contact. We normalize the values of deformation maps to $[0, 1]$ as the prediction target. After rendering the deformation map from TouchNet, then we use Taxim to render the tactile RGB image of resolution $120 \times 160 \times 3$.

Material Type	ρ	E	v	α	β
Ceramic	2.70×10^3	7.20×10^{10}	0.19	6	1×10^{-7}
Glass	2.60×10^3	6.20×10^{10}	0.20	1	1×10^{-7}
Wood	7.50×10^2	1.10×10^{10}	0.25	60	2×10^{-6}
Plastic	1.07×10^3	1.40×10^9	0.35	30	1×10^{-6}
Iron	8.00×10^3	2.10×10^{11}	0.28	5	1×10^{-7}
Polycarbonate	1.19×10^3	2.40×10^9	0.37	0.5	4×10^{-7}
Steel	7.85×10^3	2.00×10^{11}	0.29	5	3×10^{-8}

Table 1. Material parameters for audio simulation. $\rho, E, v, \alpha, \beta$ denote density, Young's Modulus, Posisson ratio, and Rayleigh damping parameters, respectively. All parameters are in SI units.

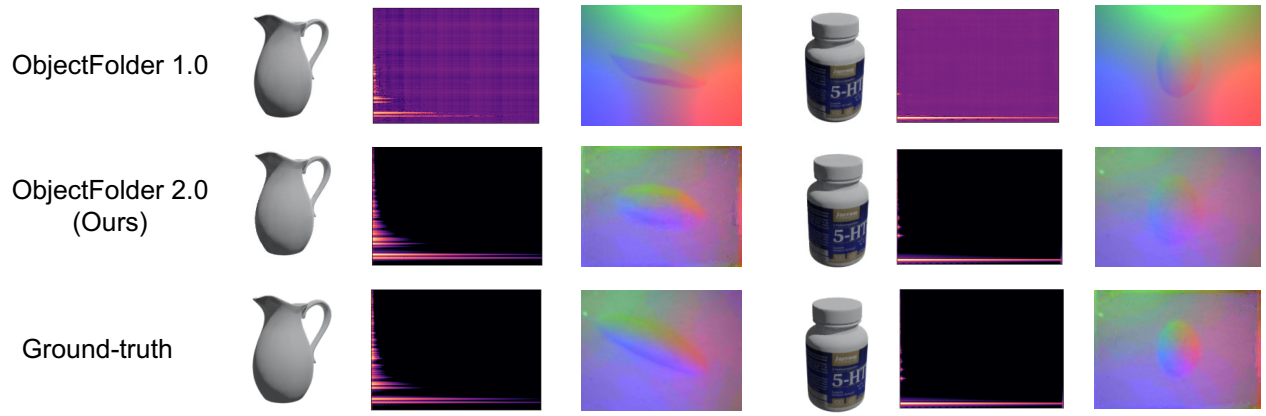


Figure 3. Comparing the visual, acoustic, and tactile data rendered from OBJECTFOLDER 1.0, OBJECTFOLDER 2.0 (Ours), and the corresponding ground-truth for two sample objects. Our implicit neural representation networks accurately encode the multisensory data for the objects that match the ground-truth well.

E. More Qualitative Examples of Mulitisensory Data

In Fig. 4 in the main paper, we have shown an example of the visual, acoustic, and tactile sensory data obtained from OBJECTFOLDER 2.0 compared with the ground-truth and the multisensory data from OBJECTFOLDER 1.0. Fig. 3 shows more examples for two objects. Our KiloOSF VisionNet renders images that match the ground-truth well while being $60\times$ faster than OBJECTFOLDER 1.0. While directly predicting audio spectrograms cannot capture the details of the modes signal and leads to artifacts in the background, our AudioNet renders audio in a much more accurate manner. For touch, to make a fair comparison, we use the TACTO [21] simulation used in OBJECTFOLDER 1.0 and the tactile readings from real-world GelSight sensors as the ground truth instead. Our TouchNet output matches well with the real tactile readings. The results demonstrate the accuracy of our implicit neural representations to encode the multisensory data for the objects.

F. Experiment Set-up Details for the Three Sim2Real Tasks

F.1. Object Scale Estimation

To perform object scale estimation, for each object we generate 500 images from different camera viewpoints and lighting conditions, 500 impact sounds with random forces for different vertexes, and 500 tactile RGB images at random surface locations from its corresponding *Object File*, respectively. For each sensory modality, we use 80% of the data for training, 10% for validation, and 10% for testing. For the vision modality, we use the ImageNet pre-trained ResNet-18 [11] network that takes an RGB image of the object as input and predicts the scale of the object. We use images of resolution 256×256 and perform random cropping to obtain images of resolution 224×224 as input to the network. After the final pooling layer, we use a fully-connected layer to map the feature vector of dimension 512 to a single value followed by a Sigmoid layer then scaled by 0.5 to predict the scale of the object. For the audio modality, we also use ResNet-18 except that we change the first layer to take a one-channel magnitude spectrogram of di-

mension 257×201 as input. The other settings are the same as the vision modality. For touch, we use the ImageNet pre-trained ResNet-18 network to extract features of dimension 512 from the tactile images. Then we forward a sequence of features extracted from 10 tactile images and each concatenated with a 128 dimension embedding of the relative positions between two consecutive touches to a 3-layer LSTM network with 256 dimensional hidden states. The final hidden state of the LSTM network is passed through two linear layers of dimension 256 and 128 to predict a scalar value representing the object scale.

For vision, to reduce the domain gap for Sim2Real, we replace the background of real-world images with white background, which is the same as the simulated images to prevent the model from overfitting to the background. For audio, because real-world recordings often contain ambient sound apart from the recorded impact sounds of the objects, during training we mix simulated impact sounds with random real-world recordings in the same environment to reduce the domain gap.

F.2. Tactile-Audio Contact Localization

We apply particle filtering [15] to localize the sequence of contact locations from which tactile readings or impact sounds are collected. For touch, we extract feature maps from an FCRN network [13], which is pre-trained for depth prediction from tactile images, after the last layer of the encoder. We directly flatten the feature map to a 81,920 dimensional feature vector. For audio, we extract the mel frequency cepstral coefficients (MFCCs) features from each 3s impact sound with number of MFCCs equal to 20 and sampling rate equal to 44,100 Hz. The obtained mel spectrogram of dimension 259×20 is flattened to a vector of dimension $5,180$ to represent the audio feature. To combine the two modalities, we concatenate the features from touch and audio. We compare these features with particles sampled from the object surfaces that represent the candidate contact locations using cosine similarities. For initialization, we randomly sample 2,500 particles from the signed distance function (SDF) that represents the object. In each iteration, we re-sample the particles based on the similarity scores. We assign larger weights to particles of larger similarity scores (smaller cosine distance). At the end of each iteration, the particles are moved according to the relative motion between two consecutive contact, and we also add Gaussian noise with the standard deviation of 2mm as the system noise.

F.3. Visuo-Tactile Shape Reconstruction

We use Point Completion Network (PCN) [22], a learning-based approach for shape completion, as a testbed for this task. For touch, the input of the PCN network is the sparse point cloud and the output is the completed dense point cloud. We sample the sparse point cloud from 32 touch positions. For each touch, a local point cloud is recovered from the tactile image from which we sample 64 points. We use 400 sparse point clouds obtained from 400×32 simulated tactile images per object for training the PCN network, and 50 each for validation and testing. During testing, we also evaluate the trained model on real tactile data. For vision, instead of using the original encoder from PCN, we replace it with a ResNet-18 network as image encoder and combine it with the PCN decoder to output a dense point cloud. Similarly, we use 400, 50, 50 images for training, validation, and testing, respectively. To combine the two modalities, we first separately train tactile-based and vision-based shape reconstruction networks, and then we concatenate the estimated point clouds from both modalities and pass it through two linear layers of dimension 1024×8 and 1024×4 to predict the final dense point cloud. We only train the last two linear layers and freeze other parts of the network.

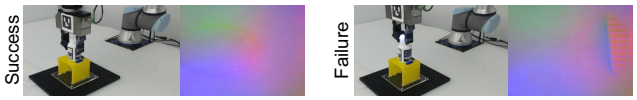


Figure 4. Examples of a successful grasp and a failed grasp.

G. Grasp Stability Prediction

We have evaluated our tactile data on three Sim2Real tasks using 13 real objects of complex shapes (Fig. 5 in the main paper), including object scale estimation, contact localization, and shape reconstruction, demonstrating the realism of our modeling. Additionally, we have performed a new challenging object manipulation task—grasp stability prediction. The goal is to predict grasp stability of a robotic arm before lifting the object based on tactile images [6, 7, 21]. We generate the tactile images from our TouchNet based on the grasp contact positions, orientations, and gel deformations, and then label each grasp as either "Success" or "Failure" based on the grasp outcome. We train a ResNet-18 binary classifier based on the touch images for the medicine bottle object. This experiment is similar to that in TACTO [21], except that we obtain touch readings by querying TouchNet. While they only test in simulation, we perform Sim2Real and test the classifier, trained only on simulated data, on real GelSight images. We obtain 72.2% accuracy., while chance is 50%. The results indicate that learning with tactile data from our dataset transfers to real-world settings. Fig. 4 shows a successful grasp and a failed grasp.

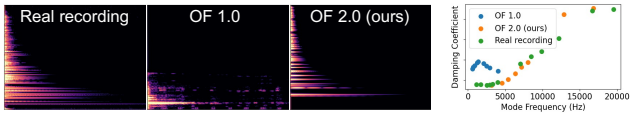


Figure 5. Examples of a successful grasp and a failed grasp.

H. Additional Analysis on Audio Simulation

Using the YCB mug object as an example, we visualize its spectrograms from OBJECTFOLDER 1.0 (OF 1.0) and OBJECTFOLDER 2.0 (OF 2.0), as well as a real recording of its impact sound in Fig. 5. Ours better matches the real recording. Furthermore, we estimate mode peaks and mode damping coefficients by finding peaks in the FFT, and then fitting a line to the log amplitude envelope over time of each mode frequency. We visualize these analytical DSP-based estimates of frequency and damping for the three respective audio clips. Our audio simulation aligns well with the real recording, much more accurately than that of OF 1.0. These comparisons as well as our Sim2Real experiments demonstrate the accuracy of our audio simulation.

References

[1] Blender - a 3d modelling and rendering package. http://www.blender.org. 1

[2] OpenGL. https://www.opengl.org. 2

[3] Pyrender. https://github.com/mmatl/pyrender. 2

[4] FEA Abaqus et al. Dassault systemes simulia corporation. 2021. 1

[5] Gaurav Bharaj, David IW Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. Computational design of metallophone contact sounds. *ToG*, 2015. 2

[6] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *RA-L*, 2018. 4

[7] Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes? In *CoRL*, 2017. 4

[8] Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. In *CoRL*, 2021. 1

[9] Ruohan Gao, Zilin Si, Yen-Yu Chang, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. Objectfolder 2.0: A multisensory object dataset for sim2real transfer. In *CVPR*, 2022. 1

[10] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[12] Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. Tetrahedral meshing in the wild. *ToG*, 2018. 1

[13] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 4

[14] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *RA-L*, 2020. 2

[15] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 1998. 4

[16] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[17] Akhil Padmanabha, Frederik Ebert, Stephen Tian, Roberto Calandra, Chelsea Finn, and Sergey Levine. Omnitact: A multi-directional high-resolution touch sensor. In *ICRA*, 2020. 2

[18] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. 2021. 2

[19] Ahmed A Shabana. *Theory of vibration*, volume 2. Springer, 1991. 2

[20] Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *arXiv preprint arXiv:2109.04027*, 2021. 2

[21] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. Tacto: A fast, flexible and open-source simulator for high-resolution vision-based tactile sensors. *arXiv preprint arXiv:2012.08456*, 2020. 3, 4

[22] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, 2018. 4