Snap-it, Tap-it, Splat-it: Tactile-Informed 3D Gaussian Splatting for Reconstructing Challenging Surfaces

Supplementary Material

A. Overview

In this appendix, we provide additional details regarding our methodology, experiments and results. Specifically, we cover our setup for data collection (Section B), the technique used for mapping tactile images (local depth maps) into 3D points (Section C), and additional results and ablations (Section D).

B. Data collection

Our methodology integrates RGB images captured by cameras with tactile images (local depth maps) acquired through robotic tactile sensing, all within a 3D Gaussian Splatting representation. This combination improves geometry reconstruction by incorporating both global information from the cameras and local details obtained through tactile sensing.

The acquisition of touch point clouds varies between simulated environments and real-world applications. In simulations, such as those involving the Shiny Blender and Glossy Synthetic datasets referenced in the main paper, we utilise the available 3D models to directly simulate local point clouds on the virtual object surfaces (Figure 7). In contrast, gathering touch point cloud in the real world necessitates to physically interact with the object through tactile sensors.

A tactile sensor [13, 14, 33] is a device mounted on the end-effector of a robot arm that transduces a surface contact area into a tactile image. This image is generated when the sensor's soft skin deforms upon touching an object, allowing an internal camera to capture the deformation. This deformation, represented as a depth map, reflects the local surface characteristics of the object. The resulting image is then converted to a 3D point cloud.

For real-world data collection, we equipped an Allegro Hand with four DigiTac tactile sensors [15] mounted on a UR5 robotic arm. We gathered 20 tactile images (5 grasps with a 4-finger robotic hand), each paired with the precise 3D pose of the sensor (in the robot frame of reference). Additionally, we collected 25 RGB images along with their corresponding camera poses (also calibrated in the robot frame) to support volume rendering. Figure 8 shows a sample of RGB images and tactile images collected on a reflective object.

C. Mapping tactile images to 3D point clouds

The conversion of tactile images to 3D point clouds is non trivial, as the local depth maps obtained with tactile sensors depends on physical characteristics and internal hardware of the tactile sensors in use. For example, while the DIGIT [13] sensor has a flat profile and produces RGB readings, sensors from the TacTip family [14] have rounded designs and outputs marker-based maps that are later preprocessed into depth maps. Therefore, directly projecting 3D point clouds from these depth maps does not always yield precise results.

In this section, we describe a solution to the shortcomings mentioned above, which draws upon the work proposed in [3, 24]. This approach involves training a model for image-to-3D point cloud mapping and can be adopted for multiple sensors, regardless of their technical specificity. A critical step in this process is the acquisition of ground truth 3D point clouds for training, which, although readily obtainable in simulations, presents significant challenges in real-world settings. Therefore, our 3D point clouds collection is performed in simulation within the environment Tactile Gym [2], which is based on the physical simulator PyBullet [4]. However, the tactile images obtained in simulation and those acquired in the real world are inherently different. Overcoming this discrepancy, known as *real-to-sim* gap, is essential for the accuracy of the conversion process. The approach to bridge real-to-sim data varies depending on the type of sensor used. For DIGIT sensors, methodologies that are discussed in Smith et al. [23, 24] can be adopted, while strategies for TacTip sensors are covered in [2, 15]. In our research we utilise the DigiTac sensor, a member of the TacTip sensor family, therefore employing available real-to-sim conversion models detailed in Lin et al. [15].

With effective real-to-sim conversion models in place, we can simulate the collection of point clouds to train a Convolutional Neural Network (CNN) for mapping tactile images to 3D point clouds. The collection of these point clouds is performed following the procedure outlined in [24]. Our training procedure begins by defining a base mesh that acts as a geometric prior for the contact surface. The CNN is then trained to predict adjustments to the vertex positions of this mesh based on tactile image inputs. This results in a deformed mesh that closely mirrors the actual geometry of the contact surface. The accurate 3D point cloud is generated by sampling points from this deformed mesh, ensuring an accurate representation of the object's touched surface. Once the CNN is adequately trained, it be-

Figure 7. Visualisation of simulated 3D point clouds (in orange) that mimic those derived from tactile images in real-world settings. The depicted objects are from the Shiny Blender and Glossy Synthetic datasets.

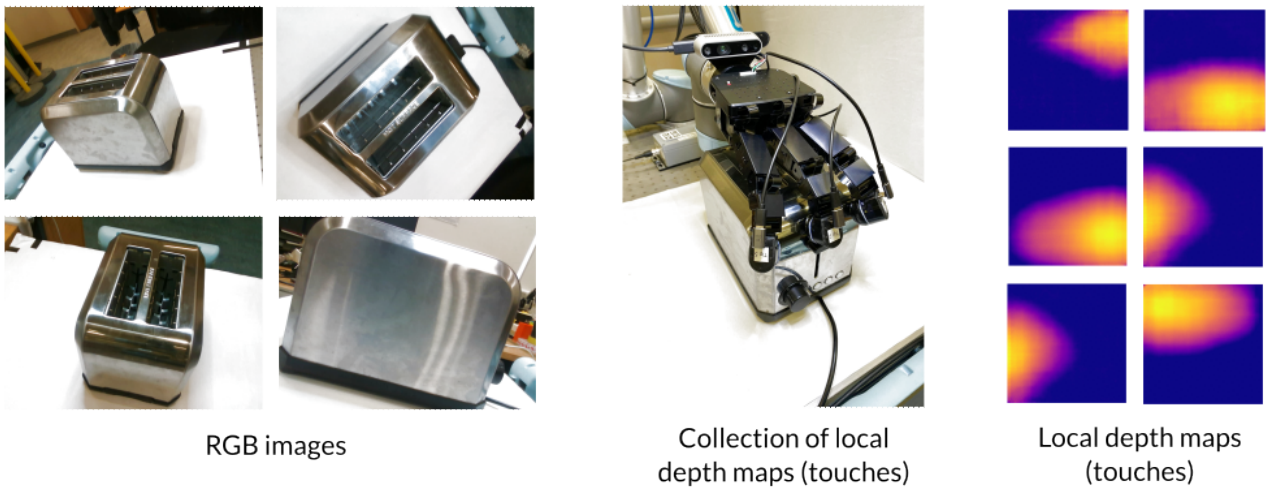

Figure 8. Data collection in the real-world. We collected RGB images and local depth maps on a reflective toaster.

comes capable of converting tactile images obtained from real-world interactions into precise 3D point clouds. We refer to [3] for additional details on this conversion mapping for TacTip sensors.

D. Additional results

D.1. Highly sparse views

In this section, we are interested to explore how our method compares against standard 3DGS on highly sparse data. Specifically, we compare its performance using from 1 to 5 views on the Glossy Synthetic dataset. Results are reported in Table 4.

We believe that 5 views strikes a nice balance, whereas we can assure that all the object is fully covered by the 5 views and it is not overwhelming for a user to capture. When using fewer views, we need to add strong inductive biases so the system can imagine unseen parts of the object, *e.g.,* pre-training a diffusion model on Objaverse renders like in [16]. We certainly believe that tactile information will help to understand unseen but touchable parts of an object. However, given the very different methodologies required for, *e.g.*, diffusion models, such an extension is out of scope for the present work.

	CD (↓)				
Views	1	2	3	4	5
3DGS	0.124	0.045	0.027	0.025	0.011
Ours	0.090	0.034	0.003	0.003	0.001

Table 4. Comparison of our method against standard 3D Gaussian Splatting (3DGS) across different view counts on the Glossy Synthetic dataset. Results show the CD from 1 to 5 views, demonstrating the effectiveness of our approach with increasing view sparsity.

Metric	Method	Glossy Synthetic [17] Angel	Bell	Cat	Horse	Luyu	Potion	Tbell	Teapot	**Avg.**
					100 Views					
CD (↓)	3DGS	0.0002	0.0132	0.0073	0.0005	0.0011	0.0143	0.0133	0.0098	0.0075
	NeRO	0.0034	0.0032	0.0044	0.0049	0.0054	0.0053	0.0035	0.0037	0.0042
	Ours[5 grasps]	0.0002	0.0068	0.0027	0.0004	0.0009	0.0038	0.0072	0.0059	0.0034
SSIM (↑)	3DGS	0.928	0.917	0.969	0.959	0.929	0.948	0.919	0.893	0.933
	NeRO	0.898	0.917	0.924	0.912	0.867	0.906	0.898	0.913	0.904
	Ours[5 grasps]	0.929	0.917	0.969	0.959	0.928	0.949	0.922	0.894	0.933
					5 Views					
CD (↓)	3DGS	0.0004	0.0220	0.0295	0.0007	0.0013	0.0122	0.0145	0.0079	0.0111
	NeRO	0.0893	0.0398	0.0230	0.1817	0.0170	0.0043	0.0859	0.0282	0.0586
	Ours[5 grasps]	0.0003	0.0082	0.0021	0.0004	0.0008	0.0013	0.0057	0.0022	0.0026
SSIM (↑)	3DGS	0.820	0.821	0.855	0.889	0.790	0.800	0.794	0.809	0.822
	NeRO	0.700	0.818	0.833	0.741	0.769	0.797	0.777	0.800	0.779
	Ours[5 grasps]	0.821	0.816	0.879	0.891	0.795	0.810	0.800	0.810	0.828
PSNR(↑)	3DGS	20.98	19.60	22.77	18.99	21.24	21.33	17.97	17.80	20.08
	NeRO	10.93	17.33	15.53	9.76	13.58	17.44	12.54	13.13	13.78
	Ours[5 grasps]	21.02	19.52	23.06	21.94	20.95	21.64	18.25	17.94	20.55

Table 5. Evaluation of the Chamfer Distance (CD) and SSIM on the Glossy Synthetic dataset [17] for individual objects. Our method is the best in recovering the object geometry from glossy surfaces in both the 100 views and 5 views scenario. In terms of photometric quality, our method outperforms the baselines in the minimal view setting.

Method	Shiny Blender, 100 views [30] Car	Coffee	Helmet	Teapot	Toaster	Avg.
			CD(↓)			
3DGS	0.0027	0.0018	0.0068	0.0003	0.0069	0.0037
3DGS + S	0.0014	0.0022	0.0024	0.0007	0.0077	0.0028
3DGS + T[5 grasps]	0.0028	0.0011	0.0043	0.0002	0.0029	0.0022
Ours[5 grasps]	0.0004	0.0017	0.0005	0.0002	0.0038	0.0013
			PSNR(↑)			
3DGS	27.40	32.81	27.56	45.48	21.10	30.87
3DGS + S	27.46	32.90	27.69	45.45	21.12	30.92
3DGS + T[5 grasps]	27.36	32.88	27.60	45.51	21.16	30.90
Ours[5 grasps]	27.43	32.91	27.48	45.10	21.19	30.82
			SSIM(↑)			
3DGS	0.930	0.972	0.950	0.997	0.896	0.949
3DGS + S	0.932	0.972	0.952	0.997	0.898	0.950
3DGS + T[5 grasps]	0.930	0.971	0.948	0.996	0.896	0.948
Ours[5 grasps]	0.932	0.972	0.951	0.997	0.899	0.950

Table 6. Evaluation of SSIM, PSNR, and CD results on the Shiny Blender dataset [30]. Our full method includes both touches and our proposed smoothness loss. Our method considerably improves the geometry reconstruction, while achieving comparable levels of image fidelity.

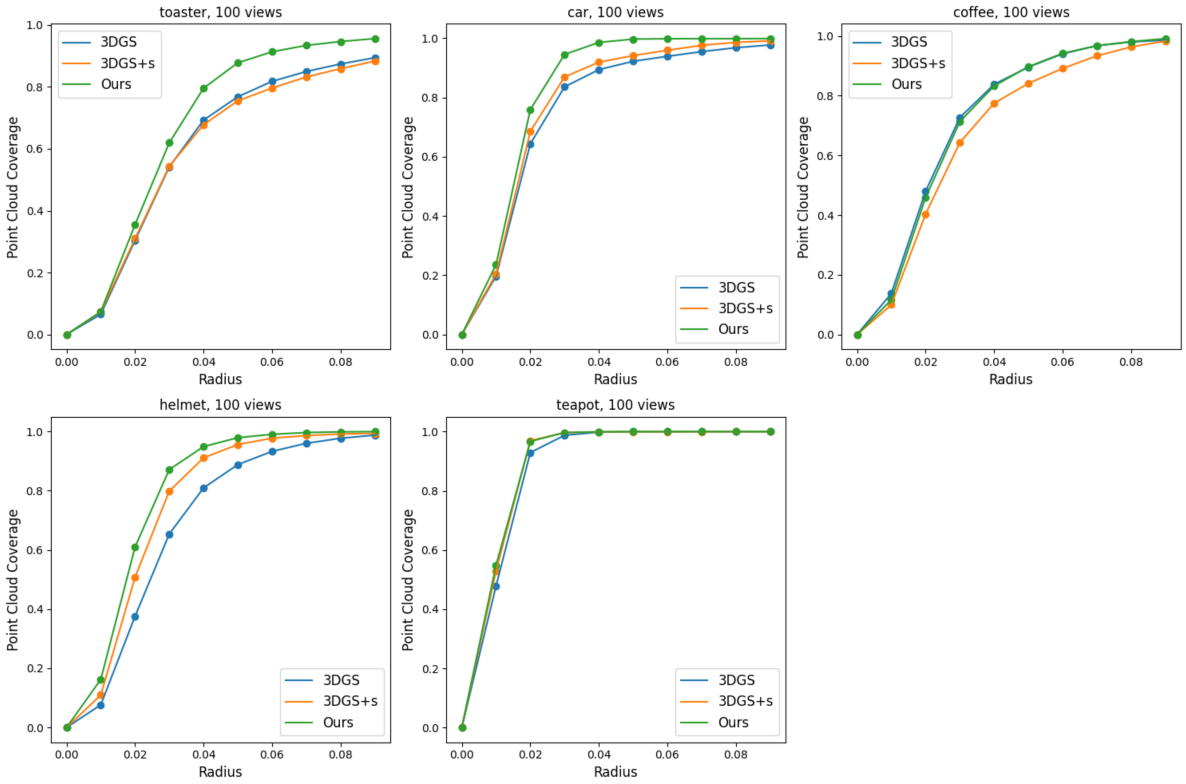

Figure 9. Point Cloud Coverage for the all the objects in the Shiny Blender dataset. Combining tactile and visual data leads to higher accuracy at lower distance thresholds

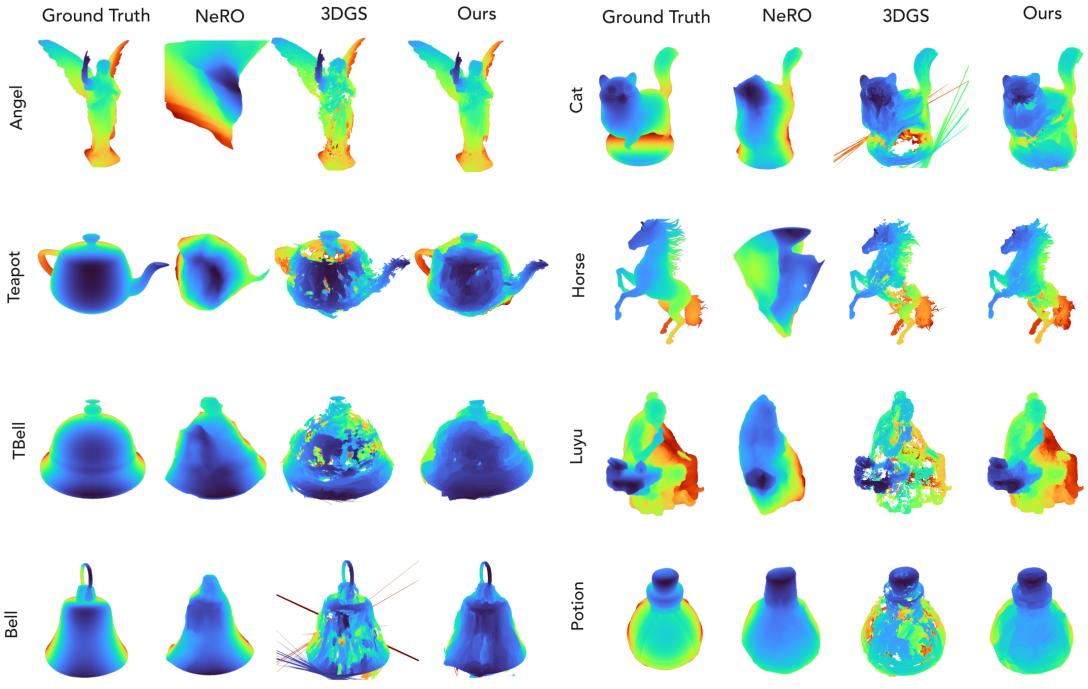

Figure 10. Surface reconstruction qualitative results using 5 training views on the full the Glossy Synthetic dataset.

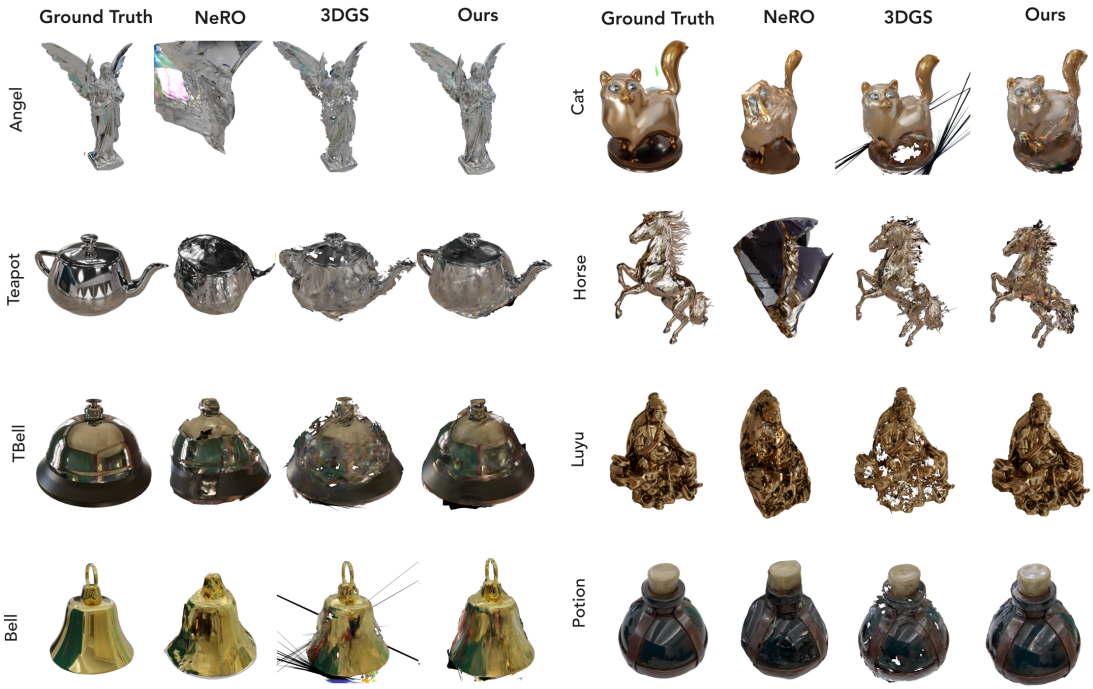

Figure 11. Novel-view synthesis qualitative results from 5 training views on the Glossy Synthetic dataset

D.2. Complete results across objects

Glossy Synthetic dataset. Table 1 in the main paper presents the average results obtained on the Glossy Synthetic dataset. This section details the results for each object in the dataset (Table 5). The results for 3DGS and our method were obtained from our experiments, while NeRO results are from the original authors. In the 100 views setting, our method outperforms the baselines in terms of CD and matches 3DGS in photometric quality. In the minimal setting view, our method surpasses the baselines in both CD and photometric quality. These findings are further supported by the qualitative results in Figure 10 (geometry reconstruction) and Figure 11 (novel-view synthesis). This underscores the main benefit of our proposed solution in applications with limited data availability.

Shiny Blender dataset. Similarly, the results for the Shiny Blender dataset in the main paper are reported as averages across the objects. Table 6 shows an ablation study on the components of our method. We report CD and SSIM for the full Shiny Blender dataset (100 views). The results indicate that combining tactile data with vision significantly improves geometry reconstruction quality, even with dense data availability, without compromising image quality.

D.3. Ablation: Point Cloud Coverage

In Section 5.3, we motivate and introduce the metric Point Cloud Coverage to complement the CD evaluation. Figure 9 displays the Point Cloud Coverage across increasing radii for each object in the Shiny Blender dataset. These curves illustrate that incorporating tactile readings improves the reconstruction quality of objects with prominent specular features, such as *toaster*, *helmet*, *car*. For objects lacking dense specular highlights, which are inherently simpler to reconstruct using standard methods, our approach performs comparably to existing techniques.