Supplementary: Meshlet Priors for 3D Mesh Reconstruction

Abhishek Badki[1,2] Orazio Gallo[1] Jan Kautz[1] Pradeep Sen[2]

[1]NVIDIA [2]University of California, Santa Barbara

1. Additional Experimentation Details

In this section we give additional details about the experiments shown in the main paper.

1.1. Generating Water-tight Meshes

We used meshlets extracted from the ShapeNet [1] dataset for training. The test dataset for mesh reconstruction was formed by selecting objects from the test set of ShapeNet dataset as well as few objects from outside the ShapeNet. However, most ShapeNet objects are not watertight. To generate water-tight meshes for our objects we used the process described by Stutz and Geiger [9]. First, the object is scaled to lie in $[-1,\ 1]^3$. Next, depth maps are rendered from 200 views and with a resolution of 1024×1024. These depth maps are used to perform TSDF fusion. We use $512 \times 512 \times 512$ volume for TSDF fusion. Finally a mesh simplification step is performed using meshlab to give us a final mesh with 50k vertices. These vertices are roughly uniform over the surface of the mesh.

1.2. Noise, Sparsity and Outliers Parameters used in Experiments

To test different approaches we designed three different settings for noise and sparsity. Given a GT object with 50k vertices, we first randomly sample a $x\%$ of the vertices to get a sparse point cloud. Following this we add a Gaussian noise of magnitude y to each point in the sparse point cloud. The three different settings used for the experiments are as follows:

- Setting 1 (S1): $x = 10\%$ and $y = 0.0150$
- Setting 2 (S2): $x = 20\%$ and $y = 0.0225$
- Setting 3 (S3): $x = 40\%$ and $y = 0.0300$

To appreciate the level of noise we provide a visualization of these noise settings in Figure 2.

1.3. Test dataset and Initialization

In Figure 1 we show all the GT objects used for the mesh reconstruction evaluation. We also show the initial mesh, $\mathcal{M}(t_0)$. Note that we used the exact same $\mathcal{M}(t_0)$ to initialize both our method and the Laplacian mesh optimization.

2. Additional Results

Figure 3 shows additional qualitative comparisons between different approaches.

We provide the numbers for each object across the different noise settings in Table 1 (Symmetric Hausdorff) and Table 2 (Chamfer-ℓ_1).

3. Additional Algorithm Details

The pseudo code for our optimization procedure to estimate a watertight mesh while enforcing meshlets priors is explained in Algorithm 1.

In our optimization procedure we update both the meshlets and the auxiliary mesh. While meshlets priors make the updates of meshlets stable, a prior is needed while updating mesh to ensure that the mesh is watertight and vertices are uniformly distributed. Use of smoothness priors or other priors for mesh would hinder our ability to reconstruct shap features. Hence we use Screened Poisson Reconstruction [5] at the end of every iteration and use the vertices and normals of globally consistent meshlets to update the mesh.

obj_name	Ours	DGP	Lap-low	Lap-high	[2]+[5]	[2]+[8]	[4]+[5]	[4]+[8]	OccNet	AtlasNet
clock_1_S3	**3.653**	17.457	11.052	11.005	122.523	15.532	11.359	15.511	29.283	24.631
clock_1_S2	**3.536**	13.955	10.953	10.891	6.235	13.111	11.333	11.858	25.27	21.75
clock_1_S1	**2.853**	13.349	11.419	11.356	3.971	10.19	11.778	15.908	25.274	21.229
chair_1_S3	**5.434**	16.978	32.646	33.899	115.056	17.746	14.692	16.042	22.796	22.476
chair_1_S2	**3.154**	12.676	30.231	32.58	112.8	13.675	14.424	13.751	12.777	19.855
chair_1_S1	**4.204**	10.124	29.066	32.861	4.012	12.236	14.455	12.642	12.651	17.271
sofa_4_S3	**5.327**	17.874	5.597	6.561	39.052	18.864	24.706	18.801	65.388	35.434
sofa_4_S2	6.13	15.129	**4.933**	6.111	16.215	14.744	26.939	16.667	33.341	33.299
sofa_4_S1	12.935	13.947	**9.957**	11.439	11.61	15.108	31.597	19.499	27.108	32.914
sofa_2_S3	**5.552**	18.691	8.772	12.991	166.4	17.276	20.451	15.399	26.107	54.051
sofa_2_S2	6.69	14.925	**5.316**	7.299	31.524	14.784	20.95	14.418	22.708	49.475
sofa_2_S1	**3.857**	10.758	5.074	6.611	13.18	13.112	24.549	14.801	16.465	40.677
sofa_3_S3	**4.272**	17.02	6.688	6.387	12.701	18.736	19.288	15.456	23.524	21.608
sofa_3_S2	**4.567**	16.181	6.169	5.986	6.57	15.084	20.977	14.355	23.436	22.014
sofa_3_S1	**3.413**	11.27	4.766	4.185	4.412	13.232	23.584	14.913	16.549	14.023
sofa_1_S3	4.513	16.788	4.485	**4.01**	10.122	17.564	27.773	16.158	22.573	23.399
sofa_1_S2	5.796	13.385	3.687	**4.92**	4.658	13.339	27.74	14.893	22.384	22.026
sofa_1_S1	5.474	10.798	5.449	**5.23**	5.2	15.536	30.814	18.724	7.518	20.873
bench_1_S3	5.142	16.187	5.887	**4.674**	153.809	16.129	17.476	15.43	30.692	23.743
bench_1_S2	4.551	14.132	3.899	**3.906**	112.483	13.554	18.607	12.978	26.192	17.842
bench_1_S1	**3.543**	10.362	4.36	4.379	4.493	10.929	20.693	12.559	20.622	15.442
guitar_1_S3	**7.381**	14.657	10.117	10.177	163.681	16.889	7.67	15.33	20.392	25.868
guitar_1_S2	**6.18**	11.316	7.614	7.65	137.846	12.287	6.419	10.392	19.465	25.338
guitar_1_S1	**3.011**	11.2	7.332	7.415	9.681	9.027	5.831	8.532	18.356	26.169
suzanne_S3	**5.719**	16.841	6.748	7.252	25.533	16.781	6.49	15.598	34.363	34.366
suzanne_S2	**4.141**	14.976	6.327	6.516	5.124	15.299	7.623	14.374	32.841	27.383
suzanne_S1	**4.797**	10.417	5.473	5.442	**4.721**	9.305	7.051	14.159	30.643	23.145
teapot_S3	**6.152**	17.25	11.014	11.161	118.128	18.79	11.768	15.414	26.124	25.139
teapot_S2	**7.314**	15.836	10.832	11.473	17.74	15.421	11.692	13.771	23.316	22.191
teapot_S1	**7.58**	11.849	10.854	11.121	6.713	10.7	11.139	13.41	24.015	18.796
armadillo_S3	**5.812**	19.349	23.069	23.48	117.21	18.891	10.438	17.449	41.609	37.975
armadillo_S2	**4.247**	16.592	23.446	23.703	8.058	14.834	12.623	12.515	40.507	36.749
armadillo_S1	**4.342**	12.021	22.962	24.023	6.943	16	14.953	14.678	39.251	35.446
bunny_S3	**3.917**	25.159	19.27	19.099	25.02	20.017	20.489	17.757	75.799	50.199
bunny_S2	**6.974**	18.543	19.11	19.198	8.336	14.737	21.744	18.383	66.464	47.761
bunny_S1	**3.967**	13.767	19.421	19.429	5.853	13.703	23.577	14.289	60.246	52.51
mobile_1_S3	4.189	16.22	4.63	**3.216**	82.184	18.032	5.012	15.388	29.668	18.336
mobile_1_S2	3.579	13.233	3.94	**3.212**	4.531	14.088	3.509	12.554	28.743	16.21
mobile_1_S1	2.869	9.153	2.979	**2.484**	3.201	10.468	5.165	10.361	28.298	16.223
speaker_1_S3	**5.033**	18.527	8.624	8.327	17.667	19.866	14.627	16.993	34.595	48.009
speaker_1_S2	**5.141**	14.938	7.281	7.328	21.282	15.363	14.223	14.691	17.407	47.535
speaker_1_S1	**4.737**	14.208	5.145	**4.72**	5.276	13.521	12.756	12.722	9.65	47.674
mailbox_1_S3	10.031	16.761	6.169	**5.413**	38.719	20.275	5.141	15.046	35.726	27.246
mailbox_1_S2	10.368	12.95	4.312	**4.874**	9.469	12.663	4.967	12.598	30.253	26.995
mailbox_1_S1	9.868	11.426	4.425	**6.325**	4.245	8.704	6.586	8.777	26.279	27.183
camera_1_S3	25.875	19.303	5.048	**5.05**	29.712	16.55	5.022	16.955	33.868	39.07
camera_1_S2	5.622	12.472	5.091	**5.331**	6.308	14.748	9.199	15.082	35.641	35.475
camera_1_S1	4.787	7.959	4.943	**4.081**	4.349	13.25	14.537	17.113	26.015	30.47
table_1_S3	**2.944**	21.648	4.165	**2.996**	30.838	18.05	14.61	16.602	46.057	26.448
table_1_S2	**3.182**	16.677	3.886	**3.124**	6.728	15.498	13.449	16.89	42.707	17.832
table_1_S1	**3.185**	8.448	3.563	**3.369**	3.934	12.077	13.865	16.576	40.015	16.728
table_2_S3	**4.791**	18.325	21.119	21.256	47.586	20.083	24.451	20.052	39.425	40.386
table_2_S2	**4.568**	15.504	21.173	21.174	28.283	13.954	21.651	15.999	38.058	43.241
table_2_S1	**5.523**	9.842	20.715	21.051	12.037	13.089	24.821	18.586	37.659	38.992
monitor_1_S3	**4.985**	19.937	**4.791**	5.151	27.943	18.517	8.319	16.546	32.03	26.32
monitor_1_S2	**5.325**	16.997	5.748	5.66	5.546	16.31	7.464	15.472	14.612	21.924
monitor_1_S1	**5.036**	8.414	5.448	5.518	5.156	13.91	5.522	15.862	9.04	21.262
lamp_1_S3	**3.934**	18.594	11.321	11.347	10.307	18.322	9.497	15.259	43.916	29.247
lamp_1_S2	**3.112**	13.952	10.25	10.115	4.465	13.1	11.141	14.075	42.604	27.942
lamp_1_S1	**4.107**	12.046	9.68	9.792	4.88	11.655	15.259	17.152	41.499	27.995
mean	**5.482**	14.655	9.974	10.255	33.871	14.921	14.741	15.069	30.497	29.363
median	**4.789**	14.791	6.507	6.931	10.2145	14.809	14.044	15.359	28.520	26.384

Table 1: Hausdorff distances between GT and estimated mesh. The distances are multiplied by a factor of 100 for better readability. We highlight in bold the best performing approach as well as other approaches that are within 0.2 error of the best approach. We show results with normals estimated with Meshlab [2] and PCPNet [4]. We reconstruct the resulting point clouds with both RILMS [8] and Screened Poisson [5]. Laplacian regularizer [7] is shown for two levels of smoothing. We also show three recent deep learning approaches: Deep Geometric Prior [10], AtlasNet [3] and OccNet [6].

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and

obj_name	Ours	DGP	Lap-low	Lap-high	[2]+[5]	[2]+[8]	[4]+[5]	[4]+[8]	OccNet	AtlasNet
clock_1_S3	**0.792**	1.658	1.199	1.071	9.294	2.562	1.265	2.378	8.748	4.26
clock_1_S2	**0.759**	1.266	1.019	0.962	1.008	1.781	1.252	1.6	5.506	3.823
clock_1_S1	**0.735**	0.908	0.931	0.943	0.827	0.995	1.301	1.374	3.951	3.576
chair_1_S3	**0.873**	1.589	1.431	1.448	7.374	2.635	1.432	2.644	7.326	3.052
chair_1_S2	**0.761**	1.204	1.21	1.392	9.841	1.871	1.247	1.821	5.169	2.358
chair_1_S1	**0.747**	0.904	0.989	1.157	0.846	1.104	1.271	1.524	3.008	1.897
sofa_4_S3	**1.14**	1.669	1.188	1.166	2.118	2.633	1.54	2.253	9.094	7.69
sofa_4_S2	1.197	1.32	**1.13**	1.163	1.226	1.567	1.757	1.883	6.557	7.39
sofa_4_S1	1.361	**1.02**	1.127	1.171	1.112	1.148	2.591	2.007	3.453	7.178
sofa_2_S3	**0.985**	1.631	1.24	1.26	6.973	2.524	1.359	2.398	7.678	7.688
sofa_2_S2	**0.967**	1.228	1.044	1.082	2.205	1.701	1.379	1.699	5.169	6.886
sofa_2_S1	**0.952**	**0.956**	**0.997**	1.043	1.257	1.213	1.785	1.727	3.256	6.304
sofa_3_S3	0.881	1.714	1.079	**0.823**	1.491	2.567	1.381	2.409	6.947	3.619
sofa_3_S2	0.832	1.272	0.922	**0.792**	0.939	1.586	1.333	1.693	5.558	2.655
sofa_3_S1	0.786	0.936	0.864	**0.759**	**0.776**	0.957	1.583	1.706	2.555	2.029
sofa_1_S3	**0.951**	1.642	1.08	**0.95**	1.369	2.551	1.672	2.356	7.21	3.583
sofa_1_S2	0.976	1.301	0.988	**0.901**	0.976	1.513	1.714	1.812	5.508	2.775
sofa_1_S1	0.937	0.991	0.925	**0.869**	0.891	1.031	2.07	1.938	2.602	2.384
bench_1_S3	**0.938**	1.547	1.199	**0.953**	11.713	2.485	1.476	2.502	7.794	3.876
bench_1_S2	**0.793**	1.202	0.916	0.809	9.116	1.827	1.514	1.788	5.702	2.697
bench_1_S1	**0.748**	0.947	0.834	0.794	0.814	1.087	1.867	1.496	3.524	1.961
guitar_1_S3	**0.973**	1.487	1.77	1.455	18.503	2.51	1.229	3.014	9.138	4.396
guitar_1_S2	**0.657**	1.161	1.046	0.837	12.261	2.004	0.8	1.945	6.986	3.166
guitar_1_S1	**0.475**	0.859	0.712	0.597	0.869	1.26	0.642	1.011	4.413	2.38
suzanne_S3	1.013	1.684	1.123	**0.93**	2.467	2.551	0.955	2.217	7.967	4.529
suzanne_S2	0.906	1.247	0.95	**0.894**	0.937	1.597	**0.873**	1.399	6.573	4.162
suzanne_S1	0.954	0.933	0.899	0.873	**0.756**	0.916	0.916	1.21	5.825	3.991
teapot_S3	**0.827**	1.72	1.184	**0.841**	9.856	2.694	1.008	2.318	6.316	3.777
teapot_S2	**0.739**	1.238	0.882	**0.744**	1.118	1.832	0.869	1.465	5.265	3.336
teapot_S1	**0.736**	0.912	0.818	**0.73**	0.711	0.977	0.876	1.206	4.66	2.902
armadillo_S3	**1.033**	1.653	1.301	1.247	7.371	2.778	1.062	2.456	9.835	6.23
armadillo_S2	**0.908**	1.208	1.071	1.132	0.999	1.735	0.972	1.604	8.848	5.931
armadillo_S1	0.962	0.914	1.007	1.1	**0.809**	1.046	1.082	1.44	7.576	5.639
bunny_S3	**1.037**	1.716	1.133	1.087	1.852	2.569	1.312	2.261	13.601	7.731
bunny_S2	1.02	1.277	1.033	1.05	**0.962**	1.453	1.283	1.529	12.126	7.482
bunny_S1	1.044	0.976	0.989	1.048	**0.839**	0.965	1.434	1.392	11.111	7.348
mobile_1_S3	**0.778**	1.72	1.123	0.795	6.66	2.835	0.94	2.332	7.237	4.035
mobile_1_S2	**0.686**	1.264	0.91	0.71	0.938	1.731	0.772	1.313	5.552	2.621
mobile_1_S1	**0.672**	0.897	0.781	**0.656**	0.738	0.925	0.717	0.903	4.136	1.832
speaker_1_S3	0.966	1.659	1.126	**0.945**	1.686	2.706	1.08	2.256	8.276	3.621
speaker_1_S2	0.92	1.244	0.987	**0.887**	1.381	1.692	1.024	1.515	4.767	3.328
speaker_1_S1	0.925	0.932	0.919	**0.868**	**0.873**	1.077	0.986	1.259	3.059	3.476
mailbox_1_S3	**0.976**	1.654	1.437	1.003	3.53	2.505	1.029	2.615	7.939	4.331
mailbox_1_S2	**0.732**	1.201	0.944	**0.733**	1.122	1.908	0.78	1.665	6.871	3.403
mailbox_1_S1	0.68	0.805	0.732	**0.648**	0.702	0.978	0.695	0.935	5.187	2.928
camera_1_S3	1.021	1.646	1.072	**0.955**	2.528	2.495	0.984	2.057	6.927	4.052
camera_1_S2	1.075	1.252	1.027	**0.953**	1.008	1.62	0.979	1.544	6.225	3.554
camera_1_S1	0.999	0.954	0.954	**0.922**	**0.93**	1.058	1.29	1.786	4.895	3.425
table_1_S3	0.825	1.697	1.032	**0.792**	1.758	2.53	1.092	2.191	9.56	3.899
table_1_S2	0.788	1.266	0.908	**0.755**	0.912	1.506	1.009	1.448	6.487	3.719
table_1_S1	0.803	0.949	0.865	**0.745**	**0.767**	0.893	1.018	1.286	3.635	3.757
table_2_S3	**0.934**	1.66	1.202	1.124	3.284	2.735	1.539	2.482	8.435	5.142
table_2_S2	**0.925**	1.279	1.1	1.08	1.665	1.622	1.474	1.91	6.948	4.359
table_2_S1	**0.926**	0.952	1.023	1.036	1.058	1.05	1.832	1.959	4.537	3.88
monitor_1_S3	1.094	1.66	1.109	1.072	2.048	2.605	**1.03**	2.033	6.693	4.368
monitor_1_S2	1.179	1.273	1.061	1.067	1.018	1.537	**0.977**	1.386	4.236	4.261
monitor_1_S1	1.13	**0.96**	1.005	1.046	**0.957**	1.017	1.033	1.338	2.354	4.017
lamp_1_S3	**0.8**	1.742	1.06	0.86	1.361	2.544	1.019	2.089	8.63	3.667
lamp_1_S2	**0.766**	1.283	0.933	0.823	0.879	1.376	0.934	1.415	5.624	3.359
lamp_1_S1	**0.772**	0.969	0.88	0.8	0.728	0.891	1.004	1.441	3.028	2.998
mean	**0.896**	1.280	1.040	0.956	2.850	1.768	1.222	1.811	6.297	4.145
median	**0.922**	1.258	1.025	0.944	1.115	1.657	1.087	1.757	6.270	3.767

Table 2: Chamfer-ℓ_1 distances between GT and estimated mesh. The distances are multiplied by a factor of 100 for better readability. We highlight in bold the best performing approach as well as other approaches that are within 0.02 error of the best approach. We show results with normals estimated with Meshlab [2] and PCPNet [4]. We reconstruct the resulting point clouds with both RILMS [8] and Screened Poisson [5]. Laplacian regularizer [7] is shown for two levels of smoothing. We also show three recent deep learning approaches: Deep Geometric Prior [10], AtlasNet [3] and OccNet [6].

Fisher Yu. ShapeNet: An information-rich 3D model repos- itory. Technical Report arXiv:1512.03012, 2015. 1

Input: Sparse and noisy point cloud PC
Output: Watertight mesh $\mathcal{M}$
`/* mesh and meshlets initialization */`
mesh, $\mathcal{M}(t_0)$, initialization (Section 4.1)
meshlets, $\{m_i(t_0)\}_{i=1:N}$, initialization by meshlets resampling (Section 4.1)
`/* outer loop */`
while $\mathcal{M}$*converges to* PC **do**
  `/* inner loop */`
  for ($n \leftarrow 0$ **to** 20 **)** {
    `/* Enforce Local Priors, Section 3.2.1 */`
    Update N meshlets $\{m_i\}_{i=1:N}$ with respect to the point cloud
    `/* Enforce Global Consistency, Section 3.2.2 */`
    while *global consistency not achieved* **do**
      Update N meshlets $\{m_i\}_{i=1:N}$ with respect to the $\mathcal{M}$
      Update $\mathcal{M}$ with respect to N meshlets $\{m_i\}_{i=1:N}$
    end
    `/* Re-meshing, Section 4.1 */`
    Estimate $\mathcal{M}$ from meshlets vertices and normals using Screened Poisson Reconstruction [5]
  }
  `/* Ensure proper coverage of meshlets on mesh */`
  Meshlets resampling (Section 4.1) on current mesh $\mathcal{M}$
end

Algorithm 1: Optimization procedure to estimate a watertight mesh while enforcing meshlets priors

[2] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: An open-source mesh processing tool. In *Eurographics Italian Chapter Conference*, 2008. 2, 3, 6

[3] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 6

[4] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. PCPNet: Learning local shape properties from raw point clouds. In *Computer Graphics Forum*, 2018. 2, 3, 6

[5] Michael Kazhdan and Hugues Hoppe. Screened Poisson surface reconstruction. *ACM Transactions on Graphics*, 2013. 1, 2, 3, 4, 6

[6] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 6

[7] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *ACM International Conference on Computer Graphics and Interactive Techniques (GRAPHITE)*, 2006. 2, 3, 6

[8] A. Cengiz Öztireli, Gael Guennebaud, and Markus Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*, 2009. 2, 3, 6

[9] David Stutz and Andreas Geiger. Learning 3D shape completion under weak supervision. *CoRR*, 2018. 1

[10] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 6

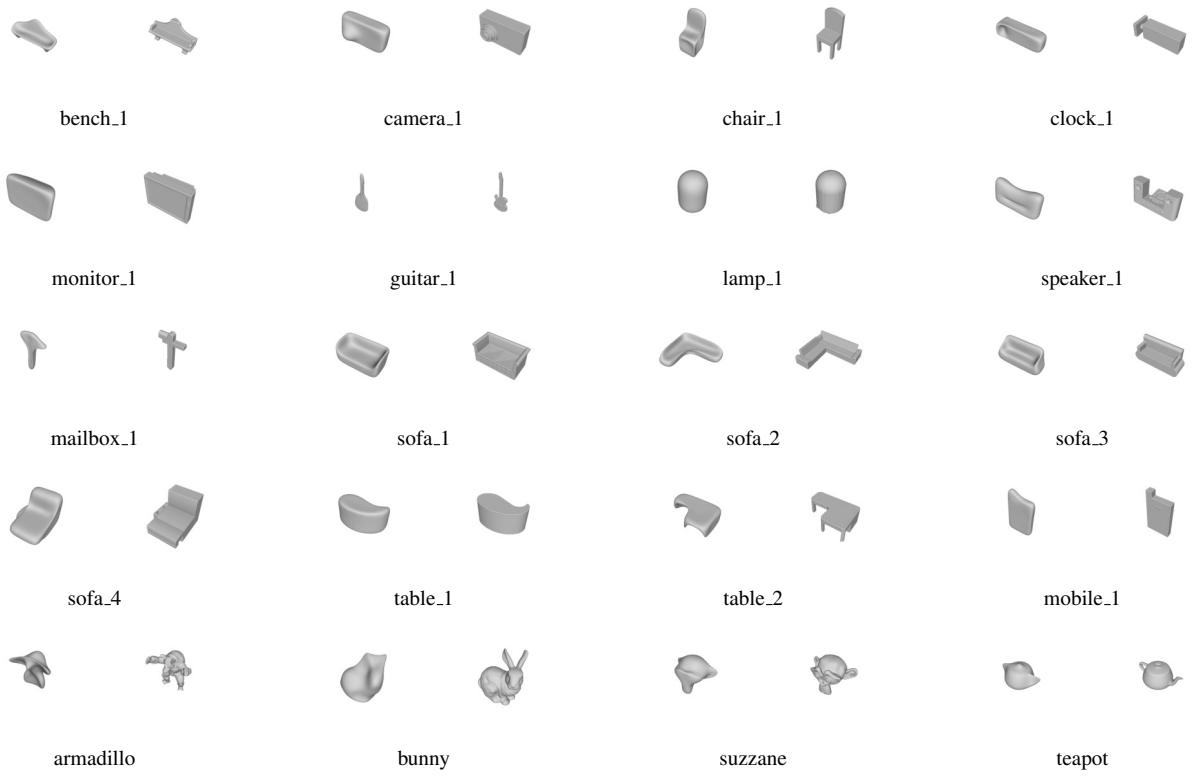

Figure 1: In this figure we show the initialization mesh $\mathcal{M}(t_0)$ and GT pair for all of the test objects. We use $\mathcal{M}(t_0)$ to initialize our method, as well as the Laplacian mesh optimization.

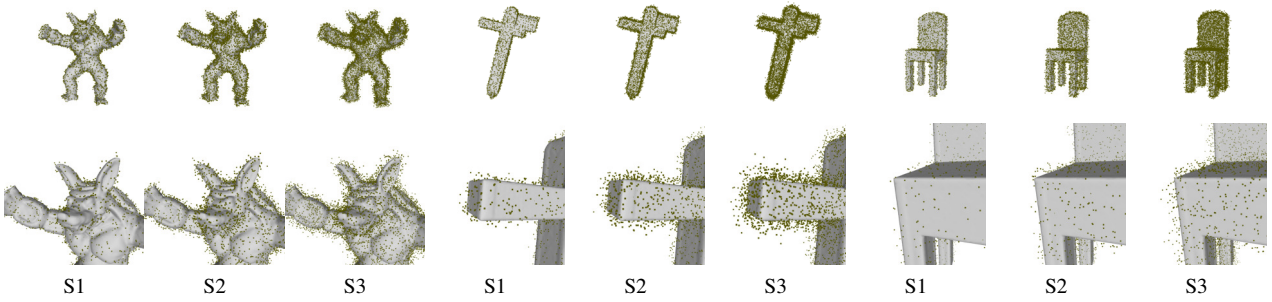

Figure 2: In this figure we show three different noise and sparsity settings used in our experiments. Setting S1 has less noise but a more sparse point cloud. Setting S3 has denser but a more noisier point cloud. These settings are designed to test the robustness of our approach to sparse and noisy observations.

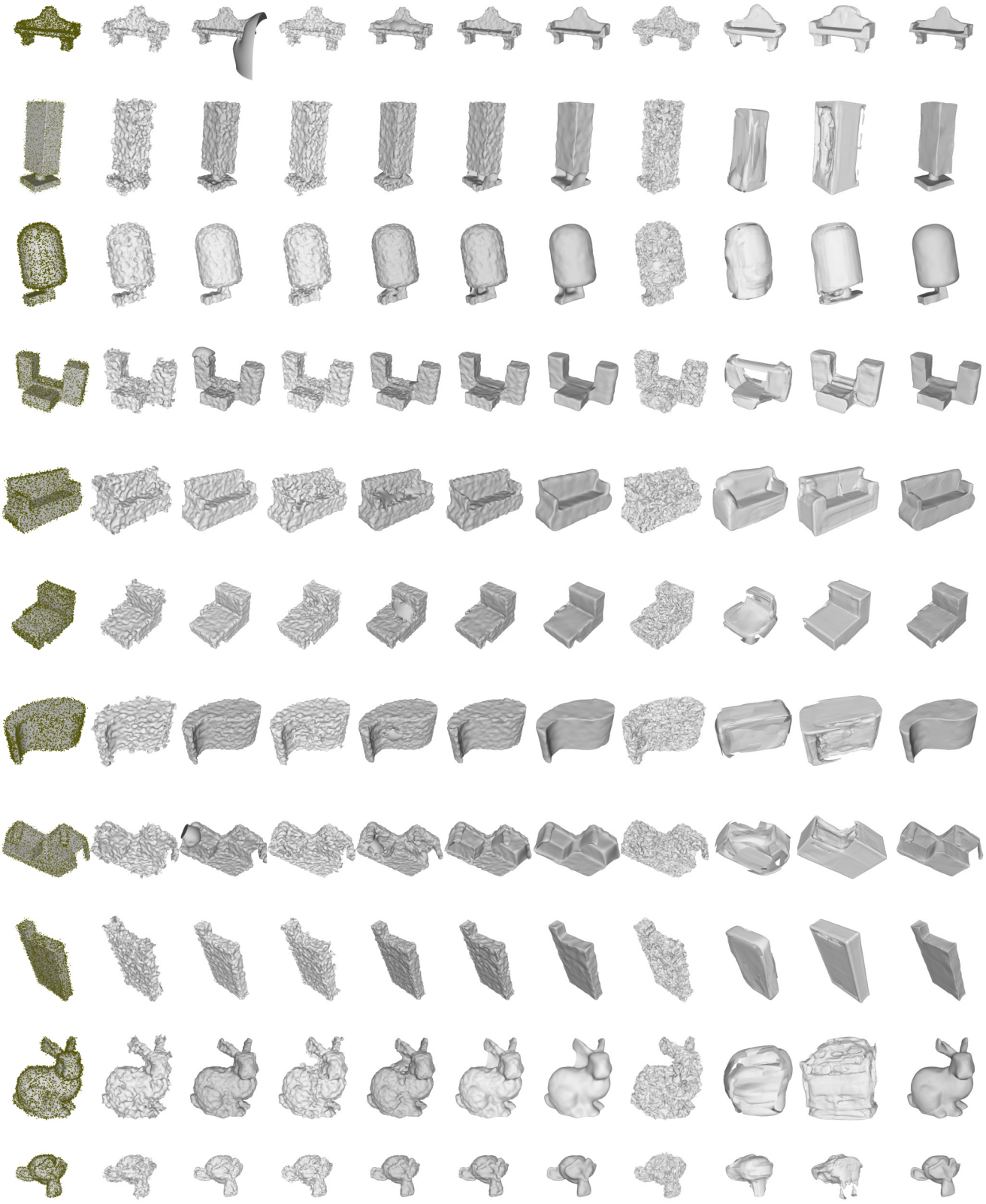

Figure 3: Qualitative comparison of several reconstruction methods and our approach. On the left is the input, the ground-truth (GT) mesh overlaid with the sparse, noisy point cloud (PC). We show results with normals estimated with Meshlab [2] and PCPNet [4]. We reconstruct the resulting point clouds with both RILMS [8] and Screened Poisson [5]. Laplacian regularizer [7] is shown for two levels of smoothing. We also show three recent deep learning approaches: Deep Geometric Prior [10], AtlasNet [3] and OccNet [6].