

Enhanced GPIS Learning Based on Local and Global Focus Areas

Zuka Murvanidze, Marc Peter Deisenroth, and Yasemin Bekiroglu

Abstract—Implicit surface learning is one of the most widely used methods for 3D surface reconstruction from raw point cloud data. Current approaches employ deep neural networks or Gaussian process models with the trade-offs across computational performance, object fidelity, and generalization capabilities. We propose a novel method based on Gaussian process regression to build implicit surfaces for 3D surface reconstruction (GPIS), which leads to better accuracy in comparison to the standard GPIS formulation. Our approach encodes local and global shape information from the data to maintain the correct topology of the underlying shape. The proposed pipeline works on dense, sparse, and noisy raw point clouds and can be parallelized to improve computational efficiency. We evaluate our approach on synthetic and real point cloud datasets including data from robot visual and tactile sensors. Results show that our approach leads to high accuracy compared to baselines.

I. INTRODUCTION

Surface reconstruction is an important step for modeling the environment in many applications varying from computer vision to robotics domain. There are various approaches proposed for reconstruction of object surfaces and geometry, following analytic [1], or data-driven [2], [3] solutions. A good surface reconstruction algorithm should handle inputs with varying sampling densities and at the same time generalize well across different object topologies. Despite the drastic proliferation of 3D scanning devices, efficient and accurate reconstruction remains an active research topic [4].

Gaussian Process Implicit Surface (GPIS) representation has been proposed [5] as a probabilistic approach to surface reconstruction. In particular, it has been shown to be efficient for different tasks in robotics: learning continuous sliding paths [6], single-finger tactile exploration to guide a sensor to high uncertainty regions [7], grasp planning in 2D using only visual



Fig. 1. Example ground truth models (the first row) and reconstructions based on our approach (the second row).

data [8], blind grasping by following surface contours for shape estimation [9], and grasping based on pre-trained systems using wrist poses [10], grasping in 3D combining visual and tactile data [11], exploring surfaces and building compact 3D representations of the environment [12]. Recent research also proved its effectiveness in real-time applications, such as robot vision, using a continuous mapping from sparse measurements [13].

Standard GPIS approaches lead to approximate reconstructions capturing general shape leaving out details. We address this issue and focus on improving reconstruction accuracy capturing more details about the underlying shape. Our approach provides an efficient, generic solution that generalizes well across the different types of inputs and fits the needs of various applications without an explicit parametrization leading to high reconstruction accuracy of implicit surface learning [14] (example reconstructions can be seen in Fig. 1). We compose a complex training dataset from sparse and dense observations of synthetic CAD models and real-world data from robot visual and tactile sensors. The goal is to highlight commonly encountered problems when dealing with the different types of raw point clouds and outline the trade-offs between reconstruction accuracy and computational speed.

We present a generic pipeline for 3D surface reconstruction extending standard GPIS approach, that can work on dense sparse, and noisy point clouds. This enables reconstructions that yield higher accuracy in comparison to the standard GPIS based approaches [15]. To achieve this, we first introduce an automated data augmentation process for a raw point cloud. We then structure the input based on local and global views of the data using an Octree-based partitioning method, where we fit a global shape function based on sampling from entire object and local shape functions from partitioned segments. During

This work was supported in part by Chalmers AI Research Center (CHAIR), in part by the Chalmers Gender Initiative for Excellence (Genie), and in part by the Project AIMCoR - AI-enhanced Mobile Manipulation Robot for Core Industrial Applications. (Corresponding author: Zuka Murvanidze.)

Zuka Murvanidze and Marc Peter Deisenroth are with the Department of Computer Science, University College London, WC1E 6BT London, U.K. (e-mail: zuka.murvanidze.17@alumni.ucl.ac.uk; m.deisenroth@ucl.ac.uk).

Yasemin Bekiroglu is with the Department of Computer Science, University College London, WC1E 6BT London, U.K., and also with the Department of Electrical Engineering, Chalmers University of Technology, SE-41296 Göteborg, Sweden (e-mail: yaseminb@chalmers.se, y.bekiroglu@ucl.ac.uk).

inference, we compare uncertainty values between local and global models per query point and use the prediction values from the model with higher confidence. This allows us to maintain the correct topology of the shape and retain accurate information about the global shape. Fidelity of the reconstruction is boosted by the information regressed by the local models, which enable reconstruction of the intricate details. We present results using both synthetic and real datasets.

The remainder of the letter is organized as follows. Section I-A discusses related works. Section II introduces our surface reconstruction approach from local and global approximations, detailing the data augmentation, and partitioning steps. Section III presents evaluations using different datasets and baseline approaches, and Section IV provides concluding remarks and suggestions for future work.

A. Related Work

A wide variety of methods are available for surface reconstruction. Some of the earliest research includes analytical approaches such as Alpha-Shape [16], Ball Pivoting [17], and Poisson Reconstruction [18] that convert point clouds into meshes. One of the most frequently used classic method is the Poisson Reconstruction, which works well with dense and noise-free data, but its performance can degenerate with sparse non-uniformly sampled, or noisy observations. Implicit surface representation is another widely used approach [19], where a regression step is applied to fit an implicit function to a set of points. Various implicit function choices have been proposed, such as wavelets [20], Fourier coefficients [21] radial-basis functions [22], and multi-scale approaches [23], [24]. More recent works extend this formulation based on various data-driven approaches, e.g. deep neural networks [2], Gaussian processes [5], [15], [25].

An emerging trend in the application of deep neural networks is to learn latent feature space representations of object classes from large datasets. Scan2Mesh [26] is an example of early work in this direction. Even though this network is robust in completing the missing parts of the input, it can produce non-smooth, coarse outputs. A slightly different approach in AtlasNet [27] represents a 3D shape as a collection of parametric surface elements and directly infers an object’s surface. Other approaches present models that learn the direct mapping from latent vector encodings to implicit surface functions [2], [28], [29], [30], [31]. Such implicit representations are more suitable for modeling objects with complex topologies and produce visually appealing smooth reconstructions. However, they can encode only a limited number of classes of objects and reconstruct the shapes from those categories. Points2Surf [3] encodes a feature representation of local patches of geometry and relies on the assumption that local regions of objects have similar properties. These models generalize well across the unseen shapes and produce high fidelity reconstructions. However, they can lead to topological noise. Furthermore, for each query point, the model evaluates the entire local patch coupled with the global encoding of the object. It has a high computational cost, which causes

long training and inference times that make it impractical for real-time applications.

A good representation of object shape should ideally allow for the following requirements: i) encode uncertainty about the shape, with uncertainty varying over different surface regions; ii) optimally combining different sources of information, e.g. tactile and visual. A probabilistic shape representation, Gaussian Process Implicit Surfaces, GPIS, [5], [32] is a good candidate to address these requirements. They have been shown to produce good surface reconstructions to identify or categorize objects [15], [33], [34].

Further extensions have been proposed to improve GPIS based representations. One such extension is the incorporation of geometric priors [25] to model expectations about the object’s shape. The method enables the reconstruction of the missing parts from the partial object observations. However, the primary challenge is to find an appropriate geometric prior for novel data, which is a non-trivial task limiting the generalization capabilities. Priors also force reconstructed meshes to align with specific requirements resulting in overly smooth and vague outputs.

However, GPs suffer from the cubic training complexity $\mathcal{O}(n^3)$ with respect to the size n of the training dataset. This limits their scalability to large training sets that we often encounter in synthetic objects or high-resolution 3D scans. Many approaches [35] are proposed to offset this limitation using various methods that fall into two main categories; making global approximations to distillate the entire data using sparse Gaussian approximations [15], [36], [37], [38], and fitting local approximations involving the division of the data into learning subspace [39], [40], [41], [42], [43], [44], [45]. For instance, GP-MPU [39], applies multi-level-partitioning and replaces polynomial functions used in the original approach [42] with Gaussian processes. However it is not tested with noisy point clouds. Other work uses overlapping local GPs [46] with the underlying hypothesis that having the subset of points captured by the neighboring patches will improve the results. However, empty spaces in reconstructed maps are observed when input data is sparse or missing. Further iterations on these models produced hierarchical [41] and skeleton-based [40] approaches, which either was not tested comprehensively with real data or relies on the assumption that input point cloud is dense.

The primary focus of our work is to increase the accuracy of reconstructed GPIS, given sparse, dense, and noisy point clouds, while maintaining adequate run-time and computational efficiency. We implement a segmentation algorithm to extract a critical training set from the input and employ powerful data augmentation techniques to maximize reconstruction fidelity. We construct a final model using global and local models derived from partitioned data which leads to higher accuracy than the baseline approaches.

II. SURFACE RECONSTRUCTION

Our method combines analytical pre-processing of a given point cloud and probabilistic learning of an implicit surface function that captures the underlying shape using Gaussian process regression. In this work, we focus on learning a model for

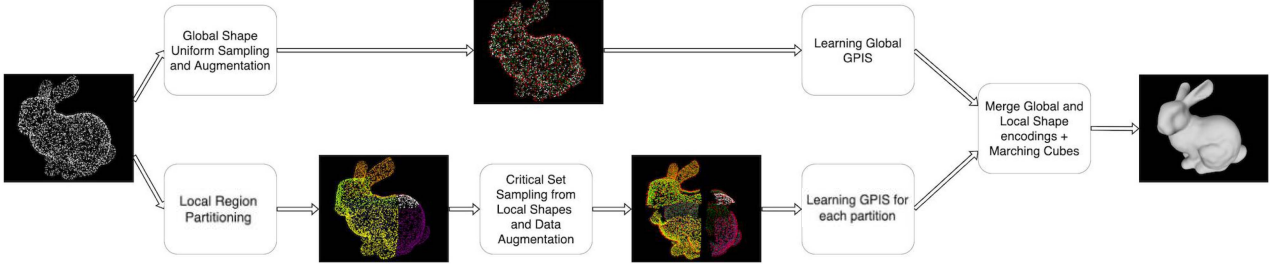


Fig. 2. System outline starting from raw point cloud input to surface reconstruction based on GP regression.

a given object. Fig. 2 illustrates multiple stages of our method, which begins with data standardization and augmentation. In the first step, we augment the data and generate additional training points that lie inside and outside of the observed point cloud. We identify the areas with more details and sample them more densely than slowly-varying regions. As a result, we obtain a critical training set, which is used in Gaussian process regression step to learn the implicit surface function. We evaluate the learned model using a cubic grid of size \mathbf{n} placed around the input object. Finally, we extract the output mesh using the Marching Cubes algorithm [47].

A. Surface Reconstruction Using Gaussian Processes

We construct an implicit surface representation based on GP regression [5] to create a model of an unknown surface. The implicit surface is defined as $f(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}$, in which $f(\mathbf{x})$ is the piece-wise function where $\mathbf{x} \in \mathbb{R}^3$ is the observed point:

$$f(\mathbf{x}) = \begin{cases} -1, & \text{if } \mathbf{x} \text{ is below the surface} \\ 0, & \text{if } \mathbf{x} \text{ is on the surface} \\ 1, & \text{if } \mathbf{x} \text{ is above the surface.} \end{cases} \quad (1)$$

A GP is a stochastic process, so that any finite collection of random variables has a multivariate normal distribution. A GP is the joint distribution of all (infinitely many) random variables and therefore it can be interpreted as a distribution over the functions $f(\cdot)$, and we write $f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot))$, where k is the covariance function of the GP.

Given a training set T with inputs $\mathbf{x}_i \in \mathbb{R}^3$ and corresponding noisy observations $y_i \in \mathbb{R}$, the relation between the function values and corresponding observations defined as $y_i = f(\mathbf{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ are independent and identically distributed noise variables, that follow a zero-mean Gaussian distribution with variance σ_n^2 . Given a finite set $\mathbf{X}' \in \mathbb{R}^{n',3}$ that comprises n' novel input points \mathbf{x}' , i.e. the points outside the training set T , the distribution of $\mathbf{f}' = [f(\mathbf{x}'_1), \dots, f(\mathbf{x}'_{n'})]^T$ is a multivariate Gaussian $p(\mathbf{f}' | \mathbf{y}, \mathbf{X}, \mathbf{X}') = \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$, whose mean and covariance are given by

$$\begin{aligned} \boldsymbol{\mu}' &= k(\mathbf{X}, \mathbf{X}') [k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y}, \\ \boldsymbol{\Sigma}' &= k(\mathbf{X}', \mathbf{X}') - k(\mathbf{X}', \mathbf{X}) [k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} k(\mathbf{X}, \mathbf{X}') \end{aligned} \quad (2)$$

respectively. Here, $k(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{n,n}$, $k(\mathbf{X}, \mathbf{X}') \in \mathbb{R}^{n,n'}$ and $k(\mathbf{X}', \mathbf{X}') \in \mathbb{R}^{n',n'}$ comprise of elements $k(\mathbf{x}_i, \mathbf{x}_j)$, $k(\mathbf{x}'_i, \mathbf{x}'_j)$

and $k(\mathbf{x}_i, \mathbf{x}'_j)$ accordingly, for i and j denoting a row and column indices. Thus $k(\mathbf{X}', \mathbf{X}) = k(\mathbf{X}, \mathbf{X}')^T$. The choice of the covariance function plays a crucial role in solving problems using Gaussian processes, because they discriminate between the functions to be selected for the inference, based on their characteristics, for instance stationarity and smoothness [48]. In our implementation, we use the radial basis function covariance function (also called squared-exponential, (3)), which is characterized by the two hyper-parameters the (function) variance σ^2 and the length scale l :

$$k^{SE}(\mathbf{x}, \mathbf{x}' | \boldsymbol{\theta}^{SE} = \{\sigma^2, l\}) = \sigma^2 \exp \left[-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right]. \quad (3)$$

Hyper-parameters are optimized using the standard training method for Gaussian processes, i.e., maximizing the marginal likelihood [48]. Empirical evaluations show that the length-scale hyper-parameter affects generalization performance across various shapes considerably, i.e. low values help capturing more details about the underlying surface features while higher values lead to smoother surfaces. The optimization procedure leads to results where we can both capture details and smooth out areas with missing points satisfactorily.

B. Data Augmentation and Processing

In addition to the points from the raw point cloud input, auxiliary training points are generated in this step to increase the accuracy of the resulting reconstructions. Ideally, the training set should contain points both inside and outside the object's surface. In order to generate these points we propose to use two approaches. The first method, radial augmentation, illustrated in Fig. 3 on the left generates spheres inside and outside the object surface to sample auxiliary points. This data augmentation method computes two distances, between the centre of the object and its furthest and closest points. These distances are then offset by a fixed constant $\epsilon = 0.035$ and used as radii to generate spheres inside and outside of the object surface, from which augmented points are sampled. A limitation of this method is that it makes a prior assumption about the modeled shape that the geometric center lies inside the surface. Consequently, they perform poorly when applied to the inputs that violate this premise such as torus.

The second technique, producing visually appealing and high-fidelity results, is per-point normal augmentation. The key idea

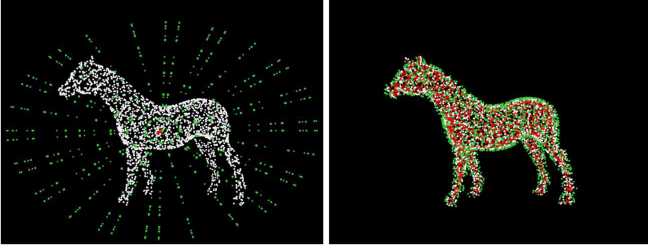


Fig. 3. Data augmentation: Radial (left) and Normal (right) based. Points inside the surface, on the surface, and outside the surface are colored in red, white, and green respectively.

is to generate two novel points for each vertex, projected in the positive and negative directions along the normal, where the projection magnitude is multiplied by a small offset value ϵ . We set this offset empirically to be $\epsilon = 0.035$ on normalized data (mean-centered, unit variance). Even though this augmentation method is very effective in terms of improving reconstruction accuracy, it triples the size of the initial training data. We often need to down-sample such datasets, so that non-parametric models, such as Gaussian processes, can efficiently handle them. We randomly down-sample the resulting augmented data preserving a set of points inside and outside the surface of the object.

If normal information is available in the data set, we apply one more step before down-sampling that is critical point selection. This selection step is performed by computing the dot product between the normals of the points in the same local neighbourhood grouped together based on their locations. The points that are selected as critical have normals that are sufficiently dissimilar to the normals of the neighbouring points. After this step, both critical and non-critical training points are sampled with the corresponding ratio of 4:1. This way we do not lose information about the less detailed areas of the object while including more points from complex areas of geometry.

C. Dataset Partitioning and Local Approximations

We apply a dataset partitioning step to create local regions where we can fit local GP models for capturing more details in our reconstructions. We use Octree data structure for subdividing the space occupied by the input point cloud into local regions, which is commonly used for compact representation of 3D volume. It also enables nearest neighbor search in logarithmic time making it further useful for segmentation and processing tasks. Fig. 4 shows different coloring of each such sub-region, where we fit the local GP models following the same pipeline as described in Fig. 2. We set the depth of the octree to 1, producing 8 equal-sized partitions. The octree depth should be increased for high accuracy reconstructions, which in turn produces more local GPs requiring additional time and computational resources for training, unless parallelized.

D. Merging Local and Global Approximations

The final step in the proposed pipeline is to combine global and local model predictions. Initially our distance voxel only

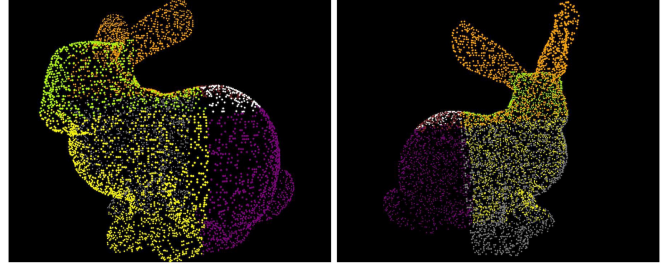


Fig. 4. Training input partitioning into local regions.

consists of global model predictions. For each partition, we pick corresponding local model and infer distance value and uncertainty value (variance) for every query point in that partition. As a result, for each point of interest we have two predictions, one from global GPIS and one from its corresponding local GPIS. We compare the uncertainty values of these predictions and select the one with the lowest variance, which improves final surfaces in comparison to only using the local predictions. We correct the global model predictions in the distance voxel where local model obtains distance measure with lower uncertainty value.

Let G_0 be a global GPIS, and G_1, \dots, G_n be local GPISs And μ_i and σ_i their corresponding means and variances. Let j denote the j^{th} query point in test sample, then for each point we select the mean value which corresponds to smaller variance:

$$\mu_{0,j} = \begin{cases} \mu_{i,j} & \text{if } \sigma_{i,j} < \sigma_{0,j} \\ \mu_{0,j} & \text{otherwise} \end{cases} \quad (4)$$

This way we merge the overall knowledge about the shape into single voxel μ_0 . This approach implicitly deals with potential discontinuities that may arise when combining different local regions together. This is because GPIS yields much higher variances at the edges of the local shapes where non-smooth (discontinued) transitions are present. In this case, global GPIS, which captures the entire geometry of the object, yields significantly lower variance, therefore the predictions from the global model are selected. This way, when two non-overlapping neighboring regions are merged, better accuracy is achieved. This volumetric grid then is fed into marching cubes algorithm to reconstruct continuous smooth surfaces with high fidelity.

III. RESULTS

In this section we present results and analysis of applying our approach for a set of reconstruction tasks, and also provide comparisons to baseline approaches. We aim to answer:

- 1) Does the data augmentation procedure help to improve GPIS reconstructions?
- 2) Do local GP estimations improve accuracy and quality of the reconstructions?
- 3) How robust is the approach to noise in the data?

We use synthetic and real datasets to evaluate our approach, eGPIS, and compare it with three baseline methods, standard GPIS approach [5], [15], [32] without the proposed data augmentation procedure, GPIS approach (denoted by csGPIS in

TABLE I

RESULTS AVERAGED ACROSS ALL OBJECTS, GPIS VS CSGPIS COMPARISON FOR ORIGINAL AND NOISY INPUTS, DISTANCES ARE BASED ON MEASUREMENTS IN MM AND TIME IN MINUTES. MINIMUM VALUES PER DISTANCE AND ELAPSED TIME FOR EACH APPROACH AND EXPERIMENT (WITH AND WITHOUT NOISE) ARE GIVEN IN BOLD

Points	GPIS (Original)				csGPIS (Original)				GPIS (Noisy)				csGPIS (Noisy)			
	CD	HD	MD	Time	CD	HD	MD	Time	CD	HD	MD	Time	CD	HD	MD	Time
2000	3.526	31.270	10.324	0.80	2.377	29.069	7.031	0.18	36.978	31.887	11.065	0.78	2.497	30.069	8.011	0.18
3000	2.932	27.371	8.958	1.59	0.840	18.761	4.198	0.41	30.288	28.316	9.836	1.76	0.637	20.616	4.537	0.70
4000	2.833	25.963	9.027	2.80	0.456	14.906	3.145	0.62	2.818	28.286	9.154	3.38	0.584	16.336	3.908	1.19
5000	2.236	21.567	7.284	6.98	0.414	13.627	3.035	0.90	2.489	24.889	7.318	8.43	0.452	15.862	3.423	1.48
6000	2.110	21.035	6.898	9.22	0.389	13.502	2.983	1.15	2.194	21.027	7.333	10.00	0.421	15.546	3.134	2.38
9000	2.035	20.996	3.910	22.12	0.338	12.160	2.625	2.51	1.783	20.165	6.913	34.23	0.392	13.160	2.625	3.67
10000	1.856	18.003	3.852	23.48	0.302	12.033	2.596	3.47	1.649	17.117	4.156	35.17	0.345	12.809	2.596	5.26

TABLE II

RESULTS PER-OBJECT: BASELINE APPROACHES P2S AND GPIS VS THE PROPOSED CSGPIS AND eGPIS. DISTANCES ARE BASED ON MEASUREMENTS IN MM. MINIMUM VALUES PER DISTANCE FOR EACH APPROACH ARE GIVEN IN BOLD, THE SECOND BEST RESULT IN RED AND THE THIRD BEST RESULT IN CYAN

Name	P2S			GPIS			csGPIS			eGPIS		
	CD	HD	MD	CD	HD	MD	CD	HD	MD	CD	HD	MD
angel	90.607	17.226	5.445	0.049	6.758	0.963	0.032	6.577	0.925	0.029	6.133	0.847
armadillo	2.920	13.377	4.427	0.094	10.291	1.748	0.077	8.961	1.460	0.074	8.399	1.439
bunny	81.713	44.280	98.086	3.361	61.306	9.532	2.594	39.887	9.867	2.288	38.212	9.739
cup	88.272	48.203	8.406	1.944	30.050	10.123	0.136	15.116	1.938	0.129	14.168	1.878
dragon	13.973	9.260	0.477	0.043	8.956	1.149	0.044	9.325	1.149	0.040	9.007	1.058
galera	14.752	42.649	0.452	5.179	34.944	17.033	0.166	10.774	2.408	0.160	10.662	2.475
hand	15.588	62.683	0.699	10.033	40.317	21.887	0.017	2.761	0.786	0.011	2.419	0.760
happy	49.724	6.630	2.386	6.056	41.859	19.176	0.055	8.310	1.282	0.051	8.476	1.232
horse	86.058	151.203	53.938	0.003	1.796	0.317	0.001	1.039	0.168	0.003	1.718	0.313
liberty	99.856	16.237	5.703	0.037	7.138	0.923	0.031	6.962	0.909	0.030	6.911	0.858
netsuke	16.580	7.742	0.452	0.119	10.334	2.071	0.116	9.323	2.119	0.116	9.033	2.121
serapis	16.452	3.225	0.547	0.290	11.757	3.599	0.273	11.333	3.539	0.274	11.219	3.558
suzanne	44.418	24.986	49.538	1.010	24.862	7.173	0.932	24.752	7.191	0.925	21.293	7.348
tortuga	12.228	2.059	0.360	0.274	3.572	1.090	0.254	15.126	3.007	0.266	13.329	2.986
Average	59.510	32.126	16.494	2.035	20.996	6.913	0.338	12.160	2.625	0.314	11.499	2.615

TABLE III

RESULTS ACROSS ALL OBJECTS USING eGPIS. DISTANCES ARE BASED ON MEASUREMENTS IN MM AND TIME IN MINUTES

Global	Local	Original				Noisy			
		CD	HD	MD	Time	CD	HD	MD	Time
5000	3000	0.345	13.188	2.708	1.47	0.362	14.094	2.814	2.74
8000	5000	0.330	12.612	2.619	6.16	0.358	13.230	2.735	7.06
9000	5000	0.286	11.858	2.522	15.97	0.316	12.249	2.624	16.51
10000	7000	0.274	11.596	2.561	18.10	0.291	12.239	2.601	19.83
11000	7000	0.206	11.118	2.531	21.54	0.284	12.192	2.590	22.94

the tables) without local enhancements using the same data augmentation introduced in the previous section, and a state of the art deep learning approach, Points2Surf [3] (P2S).

We present quantitative results in Tables I, II, and III, which detail the number of points used in the reconstruction and the distances to the ground truth using three types of distance measures: mean-square, Hausdorff, Chamfer. Table I shows the reconstruction results in terms of the distances to the ground truth for the standard GPIS approach without the new augmentation technique, and csGPIS, using the proposed data augmentation approach. We can see that the proposed augmentation approach leads the better accuracy in comparison to the baseline GPIS approach with and without noise in the data. We also observe that denser point sampling leads to higher accuracy in reconstructions at the cost of more computational time, and that given similar number of points, eGPIS leads to higher accuracy in comparison to the baseline approaches as seen from Table I and III.¹

¹Data preprocessing time including critical set segmentation and augmentation steps for csGPIS and eGPIS is around 2 to 20 seconds depending on the input point cloud. P2S preprocessing time can take minutes on a high spec GPU machine.

Table II summarizes the performance differences of a deep learning approach, Points2Surf (P2S), standard GPIS approach, GPIS without local enhancements (csGPIS) and the proposed approach (eGPIS). The proposed approach eGPIS in comparison to the baselines leads to better accuracy in general for the test objects as shown by the resulting distances to the ground truth, also supported by the averaged results in the table. Resulting GPIS models demonstrate less topological noise or errors compared to P2S. In general GPIS based results are better than the deep learning approach, consistently based on the Chamfer distance. As for the other two distance measures P2S leads to better accuracy for a few objects such as *tortuga*, *serapis*, *netsuke* where the GPIS based approaches tend to smooth out very fine and sharp details.

We visualize the reconstruction results of input point clouds for experiment objects in Fig. 5 for synthetic objects, Fig. 6 with noisy inputs for the synthetic objects, and Fig. 7 from real data collected by a robot [49], using the baseline approaches and the proposed approach. Results from Fig. 5 demonstrate the eGPIS can both capture fine details of surfaces while at the same time providing smoothness in comparison to the baselines. Fig. 8

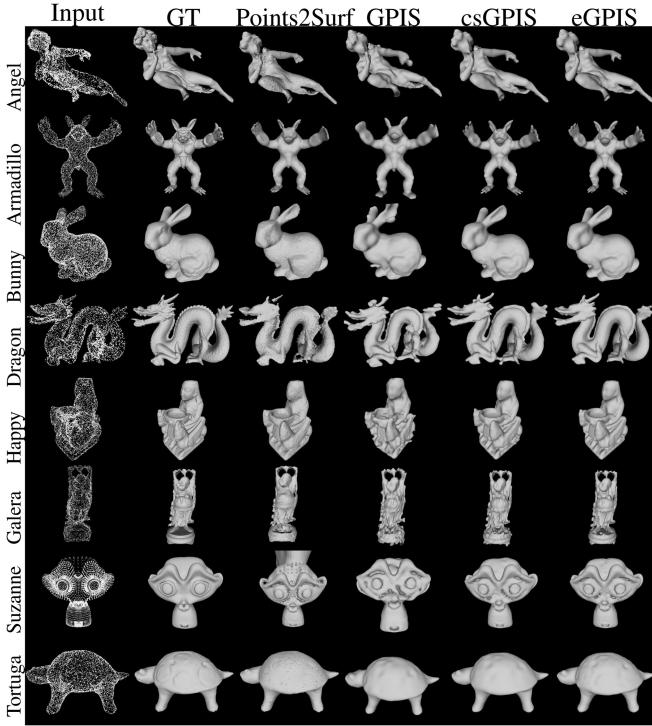


Fig. 5. Surface reconstruction results. The first column: input point cloud. GT denotes ground truth, Points2Surf is the deep learning based approach [3]. GPIS denotes the standard GP based reconstruction approach [5], [15], [32]. csGPIS denotes our approach without local enhancements, and eGPIS denotes the proposed approach with all features including augmentation and local enhancements.

TABLE IV
RESULTS FROM REAL ROBOT DATASET COMPARING CHAMFER DISTANCES TO THE GROUND TRUTH

Name	GPIS	P2S	eGPIS
box1	2.8995	12124.6591	2.1812
box2	2.4051	7708.5759	2.1053
box3	2.3028	18928.7009	1.9836
cyl1	3.2799	9470.8379	2.081
cyl2	2.9721	8729.1513	2.1625
spray1	2.4376	11309.7142	2.2386
spray2	2.5245	12296.1702	2.105
spray3	2.7742	11945.5462	2.0406
Average	2.6994	11564.1694	2.1122

provides a closer look at reconstructed surfaces from an object model that has fine details. As seen, the baseline approach P2S tend to generate details that are not available on the original surface also missing existing details such as the fingers, while eGPIS produces smooth surfaces and captures those missing details better. Fig. 6 presents the experiment results where the original input point clouds are corrupted by noise. Proposed csGPIS and eGPIS lead to better reconstructions in comparison to the other baselines and eGPIS in general produces the best results in terms of reconstruction accuracy, completeness and smoothness. These characteristics also appear in the experiments where we use real data generated using tactile and visual measurements in Fig. 7. eGPIS based reconstructions lead to complete, smooth, most accurate surfaces in comparison to the baseline approaches P2S and GPIS, yielding the smallest distances to the ground truth data as shown in Table IV.

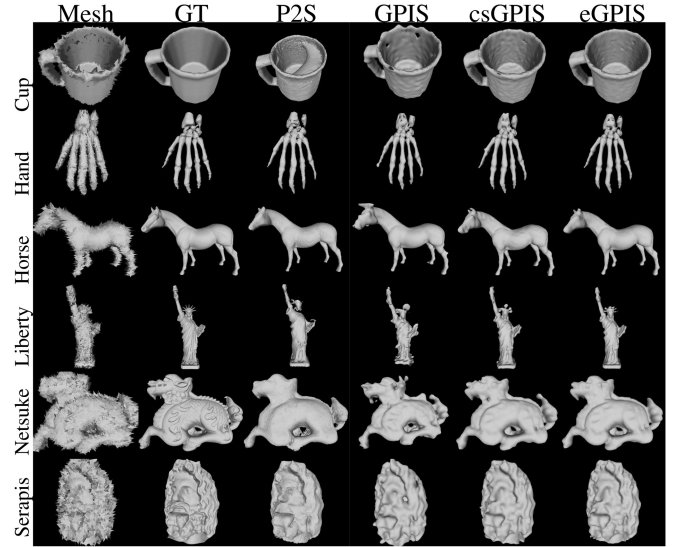


Fig. 6. Reconstructions from noisy point clouds. First Column: mesh with added Gaussian noise. GT denotes ground truth, Points2Surf is the baseline approach [3] and eGPIS is the proposed approach.

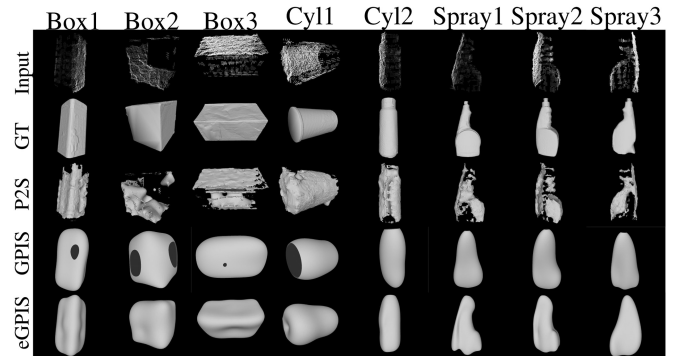


Fig. 7. Reconstructions from real robot data [49].

As for computational requirements, the deep learning approach, Points2Surf, was trained for approximately 5 days on a specialized hardware. In comparison, our approach regresses independent models per object without parallelizing or GPU acceleration. P2S does heavy pre-processing using Blender software to sample and convert point clouds into appropriate format, while our approach has more rapid and efficient pre-processing. Inference time in P2S, heavily depends on the hardware and GPU specifications, however, single CPU suffices efficient inference in our method and the partitioning together with the local GP calculations can also be easily parallelized to improve efficiency.

IV. CONCLUSION

The letter presents a 3D reconstruction approach to build implicit surface representation of a given object point cloud observation via Gaussian process regression. The Gaussian process formulation for surface reconstruction has been shown to model unknown object shape taking into account the uncertainties in observations successfully. In this letter, we further

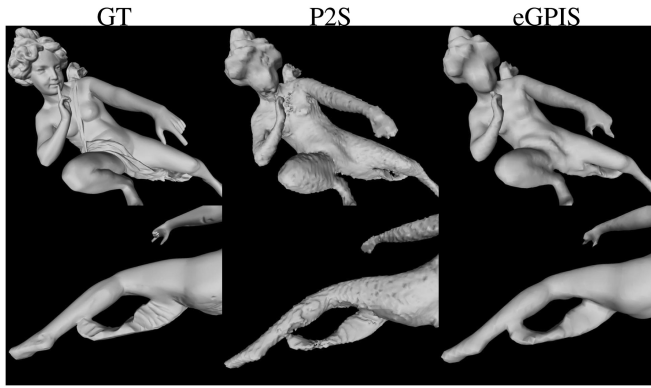


Fig. 8. Close inspection of the reconstructions by P2S and eGPIS. GT denotes ground truth reconstructions.

extend this approach and demonstrate that GPIS formulation can yield high quality reconstructions by exploiting local and global models. Results on synthetic and real-world datasets demonstrate that Gaussian processes outperform baseline approaches both in reconstruction accuracy, completeness and smoothness of the resulting surfaces.² We plan to further improve the quality of the reconstructed surfaces where fine details need to be captured. Another future work direction is to learn generic models based on training data including various objects rather than learning models per a given object.

REFERENCES

- [1] A. Pentland and S. Sclaroff, “Closed-form solutions for physically based shape modeling and recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 7, pp. 715–729, Jul. 1991.
- [2] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “DeepSDF: Learning continuous signed distance functions for shape representation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.
- [3] P. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer, “Points2surf Learning implicit surfaces from point clouds,” A. Vedaldi, H. Bischof, T. Brox, and J. M. Frahm, Eds., *Comput. Vis.*, Cham, Switzerland: Springer, pp. 108–124, 2020.
- [4] M. Berger et al., “State of the art in surface reconstruction from point clouds,” *Eurographics 2014- State Art. Rep.*, L. Sylvain and S. Michela, Eds., The Eurographics Association, pp. 161–185, 2014.
- [5] O. Williams and A. Fitzgibbon, “Gaussian process implicit surfaces,” *Gaussian Process. Pract.*, Apr. 2007. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/gaussian-process-implicit-surfaces-2/>
- [6] D. Driess, P. Englert, and M. Toussaint, “Active learning with query paths for tactile object shape exploration,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 65–72.
- [7] Z. Yi et al., “Active tactile object exploration with Gaussian processes,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4925–4930.
- [8] J. Mahler et al., “GP-GPIS-OPT: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 4919–4926.
- [9] S. Dragiev, M. Toussaint, and M. Gienger, “Gaussian process implicit surfaces for shape estimation and grasping,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2845–2850.
- [10] S. Ottenhaus, D. Renninghoff, R. Grimm, F. Ferreira, and T. Asfour, “Visuo-haptic grasping of unknown objects based on Gaussian process implicit surfaces and deep learning,” in *Proc. IEEE Int. Conf. Humanoid Robots*, 2019, pp. 402–409.
- [11] C. de Farias, N. Marturi, R. Stolkin, and Y. Bekiroglu, “Simultaneous tactile exploration and grasp refinement for unknown objects,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3349–3356, Apr. 2021.
- [12] S. Caccamo, Y. Bekiroglu, C. H. Ek, and D. Kragic, “Active exploration using gaussian random fields and Gaussian process implicit surfaces,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot Syst.*, 2016, pp. 582–589.
- [13] B. Lee, C. Zhang, Z. Huang, and D. D. Lee, “Online continuous mapping using Gaussian process implicit surfaces,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 6884–6890.
- [14] G. Turk and J. F. O’Brien, “Variational implicit surfaces,” Georgia Inst. Technol., 1999.
- [15] G. Z. Gandler, C. H. Ek, M. Björkman, R. Stolkin, and Y. Bekiroglu, “Object shape estimation and modeling, based on sparse Gaussian process implicit surfaces, combining visual data and tactile exploration,” *Robot. Auton. Syst.*, vol. 126, 2020, Art. no. 103433.
- [16] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, “On the shape of a set of points in the plane,” *IEEE Trans. Inf. Theory*, vol. 29, no. 4, pp. 551–559, Jul. 1983.
- [17] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, “The ball-pivoting algorithm for surface reconstruction,” *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 4, pp. 349–359, Oct./Dec. 1999.
- [18] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Proc. Fourth Eurographics Symp. Geometry Process.*, Goslar, DEU: Eurographics Association, 2006, pp. 61–70.
- [19] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface reconstruction from unorganized points,” *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, 1992.
- [20] J. Manson, G. Petrova, and S. Schaefer, “Streaming surface reconstruction using wavelets,” *Comput. Graph. Forum*, vol. 27, pp. 1411–1420, 2008.
- [21] M. Kazhdan, “Reconstruction of solid models from oriented point sets,” in *Proc. Eurographics Symp. Geometry Process.*, 2005, pp. 73–es.
- [22] Y. Ohtake, A. Belyaev, and H.-P. Seidel, “3D scattered data interpolation and approximation with multilevel compactly supported RBFs,” *Graphical Models*, vol. 67, no. 3, pp. 150–165, 2005.
- [23] Y. Ohtake, A. Belyaev, and H. P. Seidel, “A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions,” in *Proc. IEEE Shape Model. Int.*, 2003, pp. 153–161.
- [24] Y. Nagai, Y. Ohtake, and H. Suzuki, “Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction,” *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1339–1348, 2009.
- [25] W. Martens, Y. Poffet, P. Soria, R. RamónFitch, and S. Sukkarieh, “Geometric priors for Gaussian process implicit surfaces,” *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 373–380, Apr. 2017.
- [26] A. Dai and M. Niessner, “Scan2Mesh: From unstructured range scans to 3D meshes,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5574–5583.
- [27] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, “A papier-mâché approach to learning 3D surface generation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 216–224.
- [28] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3D reconstruction in function space,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4455–4465.
- [29] J. Mu, W. Qiu, A. Kortylewski, A. Yuille, N. Vasconcelos, and X. Wang, “A-SDF: Learning disentangled signed distance functions for articulated shape representation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 13001–13011.
- [30] V. Sitzmann, E. Chan, R. Tucker, N. Snavely, and G. Wetzstein, “MetaSDF: Meta-learning signed distance functions,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 10136–10147.
- [31] K. Gupta, S. Jabbireddy, K. Shah, A. Shrivastava, and M. Zwicker, “Improved modeling of 3D shapes with multi-view depth maps,” in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 71–80.
- [32] S. Dragiev, M. Toussaint, and M. Gienger, “Gaussian process implicit surfaces for shape estimation and grasping,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2845–2850.
- [33] M. Björkman, Y. Bekiroglu, V. Högman, and D. Kragic, “Enhancing visual perception of shape through tactile glances,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3180–3186.
- [34] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, “Active learning with Gaussian processes for object categorization,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [35] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, “When Gaussian process meets Big Data: A review of scalable GPs,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4405–4423, Nov. 2020.
- [36] M. K. Titsias, “Variational learning of inducing variables in sparse Gaussian processes,” in *Proc. Artif. Intell. Statist.*, 2009, pp. 567–574.

²The source code can be accessed via <https://github.com/Zuka98/eGPIS>.

- [37] J. Quiñero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, no. 12, pp. 1939–1959, 2005.
- [38] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable variational Gaussian process classification," in *Proc. Artif. Statist. Mach. Learn.*, 2015, pp. 351–360.
- [39] M. G. López, B. Mederos, and O. Dalmau, "GP-MPU method for implicit surface reconstruction," in *Proc. Hum.-Inspired Comput. Appl.*, 2014, pp. 269–280.
- [40] L. Wu, R. Falque, V. Perez-Puchalt, L. Liu, N. Pietron, and T. Vidal-Calleja, "Skeleton-based conditionally independent Gaussian process implicit surfaces for fusion in sparse to dense 3D reconstruction," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1532–1539, Apr. 2020.
- [41] S. Kim and J. Kim, "Hierarchical Gaussian processes for robust and accurate map building," in *Proc. Australas. Conf. Robot. Automat.*, 2015, pp. 117–124.
- [42] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, "Multi-level partition of unity implicits," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 463–470, 2003.
- [43] V. Tresp, "A Bayesian committee machine," *Neural Comput.*, vol. 12, no. 11, pp. 2719–2741, 2000.
- [44] Y. Cao and D. J. Fleet, "Transductive log opinion pool of Gaussian process experts," 2015, *arXiv:1511.07551*.
- [45] S. Cohen, R. Mubvha, T. Marwala, and M. P. Deisenroth, "Healing products of Gaussian process experts," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2068–2077.
- [46] S. Kim and J. Kim, "Continuous occupancy maps using overlapping local Gaussian processes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4709–4714.
- [47] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, "Efficient implementation of marching cubes' cases with topological guarantees," *J. Graph. Tools*, vol. 8, no. 2, pp. 1–15, 2003.
- [48] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [49] Y. Bekiroglu, M. Bjorkman, G. Z. Gandler, J. Exner, C. H. Ek, and D. Kragic, "Visual and tactile 3D point cloud data from real robots for shape modeling and completion," *Data Brief*, vol. 30, 2020, Art. no. 105335.