The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLIX-B2-2026
XXV ISPRS Congress 2026 “From Imagery to Understanding”, Commission II, 4–11 July 2026, Toronto, Canada

Shape Representation using Gaussian Process mixture models

Panagiotis Sapoutzoglou, George Terzakis, Georgios Floros, Maria Pateraki

Laboratory of Photogrammetry, School of Rural, Surveying and Geoinformatics Engineering,
National Technical University of Athens, Greece
(psapoutzoglou, gterzakis, gfloros, mpateraki)@mail.ntua.gr

Keywords: 3D shape representation, Gaussian Processes, surface modeling, probabilistic reconstruction.

Abstract

Traditional explicit 3D representations, such as point clouds and meshes, demand significant storage to capture fine geometric details and require complex indexing systems for surface lookups, making functional representations an efficient, compact, and continuous alternative. In this work, we propose a novel, object-specific functional shape representation that models surface geometry with Gaussian Process (GP) mixture models. Rather than relying on computationally heavy neural architectures, our method is lightweight, leveraging GPs to learn continuous directional distance fields from sparsely sampled point clouds. We capture complex topologies by anchoring local GP priors at strategic reference points, which can be flexibly extracted using any structural decomposition method (e.g. skeletonization, distance-based clustering). Extensive evaluations on the ShapeNetCore and IndustryShapes datasets demonstrate that our method can efficiently and accurately represent complex geometries.

1. Introduction

Shape representation constitutes a fundamental problem in a variety of fields such as 3D computer vision, photogrammetry, and graphics. Compact and accurate representations of 3D shapes are essential for tasks ranging from object modeling, reconstruction, and 6D pose estimation to the creation of digital twins, large-scale urban modeling, and heritage documentation. In these photogrammetric workflows, the transition from raw sensor data to structured surfaces is often a critical bottleneck. Standard outputs, which typically consist of massive, unstructured point clouds, suffer from noise, occlusion, and varying point densities, creating a burgeoning need for representations that bridge the gap between discrete data and continuous geometry. While explicit geometric representations (e.g., point clouds, meshes, and voxels) can capture fine levels of detail, they lack storage parsimony, leading to high memory costs and requiring complex indexing systems for quick surface lookups. On the other hand, functional models (implicit representations) are typically parametric, highly storage-efficient, and adept at learning complex topologies. Consequently, a line of research represents surfaces implicitly, either as a continuous volumetric field or as the decision boundary of a classifier. Other approaches use neural rendering to learn the object representation, optimizing a neural network, a sparse grid of spherical harmonics or an unstructured set of 3D Gaussians. Local kernel functions emerged as a building block for implicit shape representations. Overall, neural shape representations bear significant benefits, such as generalization and storage efficiency. These models are optimized to capture the statistical characteristics of an entire object class, facilitating tasks like shape completion, rather than reconstructing the high fidelity, idiosyncratic details of a single, specific geometry.

In this work we propose an object-specific implicit representation: Functional modeling of surface geometry using Gaussian Processes (GPs). GPs are prominent tools for non-parametric regression (Rasmussen, 2004). In contrast to neural models, our method leverages the ability of GPs to model continuous functions from irregularly sparse sampled data and apply this concept in the context of a probabilistic model that learns the shape of an object as the mixture of multiple directional distance fields anchored at reference points specially placed in the object’s skeletal outline. Unlike methods that utilize directional distance fields (DDFs) along with a neural architecture (Tretschk et al., 2020, Aumentado-Armstrong et al., 2022, Feng et al., 2022, Yenamandra et al., 2024), we abstain from using any neural component. Our GP approach draws from the work by Has (Has, 2023) on classifier mixtures and exploits it for shape representation. The resulting mixture model ensures geometric continuity and sparsity, capturing finer shape detail while avoiding the heavy training burden associated with deep implicit methods. The contributions of our method are two-fold.

1. We present a novel, functional shape representation as a mixture of GP based priors. This representation is “lightweight” as it completely abstains from deep neural components while operating solely on sparsely sampled surface points.

2. We evaluate the representation capabilities of the proposed model using both synthetic and real data and compare it against state-of-the-art competitors.

The source code is available at https://github.com/POSE-Lab/GP-mixture-shape-representation.

2. Related Work

2.1 Shape representation

Existing approaches for representing 3D geometry can be broadly divided into distinct paradigms based on how they mathematically model an object’s surface (Cremers, 2015). Point-clouds, meshes, and voxels are the most popular *explicit* representations able to capture arbitrary amounts of detail in the objects’ shapes. They are used either as a standalone model or in conjunction with an encoder-decoder architecture (Achlioptas et al., 2017, Wang et al., 2018, Brock et al., 2016). Typical shortcomings of

these representations include increased storage requirements for higher detail levels, and the need for special lookup structures to support operations such as nearest neighbor queries.

On the other hand, *implicit* representations are typically parametric functional models that are more storage-efficient and have proven to be adept at learning complex topologies (Farshian et al., 2023). A line of research attempts to implicitly represent the object's surface, either as a continuous volumetric field (Park et al., 2019a) or as the decision boundary of a classifier (Mescheder et al., 2019). Other approaches use neural rendering to learn the object representation, optimizing a neural network (Mildenhall et al., 2021), a sparse grid of spherical harmonics (Fridovich-Keil et al., 2022) or an unstructured set of 3D Gaussians (Kerbl et al., 2023). Local kernel functions (Williams et al., 2021, Huang et al., 2023) have also been used as a building block for implicit shape representations, focusing on the problem of surface reconstruction from sparse point clouds. Several extensions have been proposed, including the use of hierarchical structures (Paschalidou et al., 2020) and the deformation of the underlying model (Zheng et al., 2021). Overall, neural shape representations bear significant benefits, such as generalization, continuity in the input domain and storage efficiency. However, they are not purposed to reproduce details of specific shapes but rather as more general prediction frameworks. As such, they require swaths of data for training, rendering them unappealing for generating templates. On the contrary, switching to primitive representations has its own drawbacks such as increased storage to level of detail in the shape or computationally intensive lookup queries (Farshian et al., 2023, Liao et al., 2018). Although Gaussian processes are prominent tools for non-parametric regression (Titsias, 2009, Williams and Rasmussen, 2006), to the best of our knowledge, they have not been utilized as a primary functional representation for 3D shapes. The closest related work by Dragiev et al. (Dragiev et al., 2011) employs them specifically to refine shape uncertainty by integrating sensor data incrementally. More broadly, GPs have been extensively used for pattern learning from sparse data. Recently, neural GP ensembles (Damianou and Lawrence, 2013) and GP-inspired architectures (Garnelo et al., 2018, Wilson et al., 2016) have been proposed, boasting superior performance on tasks such as orientation estimation from face images or digit magnitude prediction, rather than full surface modeling.

2.2 Directional distance representation

Directional distance representation methods have a long history in computer vision (Rosenfeld and Pfaltz, 1968). Aumentado-Armstrong et al. (2022) exploits DDFs as a 3D representation framework. For a given shape it learns a field that maps any position and orientation to visibility and distance, allowing differentiable rendering within an implicit architecture. Rather than modeling a volumetric field (as in NeRFs) or an explicit mesh, DDFs use a continuous field of distances around objects, facilitating their use in neural networks for 3D scene reconstruction and other inverse graphics tasks. The paper outlines some notable limitations of the DDF approach, which relate to a) rendering complexity as they demand multiple network evaluations per pixel, making real-time applications challenging, b) requirement of high-quality depth and visibility data, and c) handling occlusions as DDFs require fine control over surface visibility, potentially leading to inaccuracies. Feng et al. (2022) leverages an implicit shape global representation, which models the object from the ray-based perspective, rather than 3D-point-based, demonstrating improved efficiency and quality over standard approaches like SDF (Park et al., 2019a). The shape is represented by a single network without any spatial partitions as in PatchNets (Tretschk et al., 2020). Though due to its dependency on known views for training it may require additional modifications to produce reliable outputs, while the evaluation cost increases in high-resolution settings and certain self-intersecting geometries remain challenging. The method of (Yenamandra et al., 2024) also utilizes a DDF representation and combines it with the SDF to accelerate 3D shape reconstruction tasks, enabling rendering with only a single evaluation per ray compared to traditional SDF-based methods that require multiple evaluations per ray for rendering. The method is optimized for speed with a simpler DDF model which limits its ability to handle occlusion and discontinuities, while (Aumentado-Armstrong et al., 2022) focuses on handling discontinuities more accurately but with additional computational overhead. In comparison to the above methods that focus on the task of differentiable rendering in inverse graphics tasks our GP-based approach aims to handle complex shapes with multiple surface intersections by distributing reference points within the object towards a method-independent pose confidence framework for real-time applications. Thus each method's approach reflects its intended application.

3. Method

We construct our shape representation by learning continuous directional distance fields (DDFs) from sparsely sampled point clouds. Unlike traditional methods that rely on neural networks, our approach leverages Gaussian Processes (GPs) to model these fields as a functional representation. Our method is summarized in Fig. 1. The shape is initially partitioned into regions and subsequently, special-purpose reference points are computed via skeletonization or distance-based clustering. The 3D points belonging to each region are then encoded as a directional distance fields along in spherical coordinates with respect to a chosen reference point. The encoding of distance fields is achieved by training local GP models that comprise our GP mixture model.

3.1 Parametrization of Directional Distance Fields

We construct our shape representations by learning DDFs on spherical coordinate domains affixed to suitable reference points in the object's local coordinate frame. Consider a reference point $\boldsymbol{C}$ in the object's local frame and a given point $\boldsymbol{P}$ on the object's surface. We parameterize $\boldsymbol{r} = \boldsymbol{P} - \boldsymbol{C}$ in terms of its length and bearing (see Fig. 2) as follows:

$$\boldsymbol{r} = \boldsymbol{P} - \boldsymbol{C} \coloneqq (\phi, \theta, d)\,, \tag{1}$$

where ϕ, θ are the spherical coordinates of the bearing vector $\boldsymbol{r}/\|\boldsymbol{r}\|$ and $d = \|\boldsymbol{r}\|$ is the Euclidean norm of $\boldsymbol{r}$. Thus, if $\boldsymbol{u}(\phi, \theta) \in \mathbb{R}^3$ is the bearing vector defined by the spherical coordinates ϕ and θ, we may obtain $\boldsymbol{P}$ as,

$$\boldsymbol{P} = \boldsymbol{r} + \boldsymbol{C} = d \cdot \boldsymbol{u}(\phi, \theta) + \boldsymbol{C}. \tag{2}$$

Thus, for a given reference point, $\boldsymbol{C}$, in the interior of an object represented by a closed surface, we may represent all or part of the surface as a collection of distances along every possible direction from $\boldsymbol{C}$.

3.2 Gaussian Processes for distance prediction

Consider an object with an exterior surface that comprises points reachable by means of ray-casting from a point, $\boldsymbol{C}$, in the interior

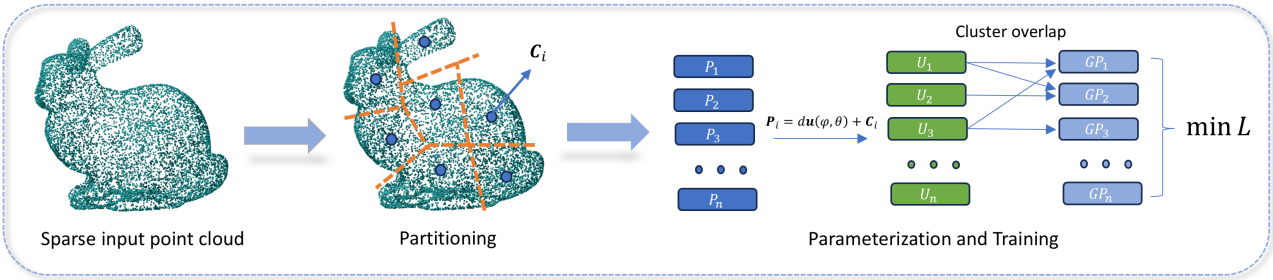


Figure 1. We partition a sparse point cloud using distance-based clustering and extract reference points (Sec. 3.1). Each 3D point P_i is parameterized as a bearing vector in spherical coordinates U_i relative to its reference point C_i. The directions, with distances as targets, serve as inputs to a Gaussian Process (GP_i) (Sec. 3.2). The GP mixture model forms our shape representation.

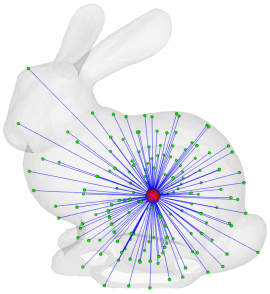


Figure 2. Spherical directional distance field (blue rays) centered at reference point $\boldsymbol{C}$ (red).

of the object. To fit this surface to a function, we model distances from $\boldsymbol{C}$ to the surface along a given direction $\boldsymbol{u}(\phi, \theta)$, as a Gaussian process (Rasmussen, 2004, Bishop, 2006, Pérez-Cruz et al., 2013),

$$d \sim \mathcal{GP}\left(0, k\right), \quad \text{s.t.} \;\; k : \left([0, \pi] \times [0, 2\pi]\right)^2 \to \mathbb{R}, \tag{3}$$

where k is the so-called covariance function mapping any two direction parameter vectors $\boldsymbol{\psi} = (\phi, \theta)$ and $\boldsymbol{\psi}' = (\phi', \theta')$ to a real number, such that the Gram matrix $\boldsymbol{K} = [k(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j)]$ is positive semi-definite (PSD) for any finite collection $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n\}$ of direction parameter vectors (Rasmussen, 2004). Different choices for the kernel k give rise to different GPs. The Rational Quadratic (RQ) kernel, known for its capacity to regulate the extent of multi-scale behavior, stands out as a more adaptive option (see Sec. 3.6) when compared to the Radial Basis Function (RBF) kernel (Shawe-Taylor and Cristianini, 2004, Williams and Rasmussen, 2006) :

$$k_{\mathrm{RQ}}(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j) = \left(1 + \frac{d_{ij}^2}{2\alpha l^2}\right)^{-\alpha}. \tag{4}$$

In this context $l > 0, \alpha > 0$ are hyperparameters and d_{ij} denotes the Euclidean distance, $\|\boldsymbol{\psi}_i - \boldsymbol{\psi}_j\|$, between the orientation parameters. Perhaps a more suitable choice for d_{ij} would be the geodesic distance between the bearing vectors $\boldsymbol{u}(\boldsymbol{\psi}_i), \boldsymbol{u}(\boldsymbol{\psi}_j)$, but that would lead to a non-PSD kernel function as shown in (Jayasumana et al., 2015), to which a close valid alternative would be the Euclidean distance, $\|\boldsymbol{u}(\boldsymbol{\psi}_i) - \boldsymbol{u}(\boldsymbol{\psi}_j)\|$.

For a sufficiently sized training set of n 3D points on the surface of the object, we may train the GP model to explicitly predict surface shape in terms of distance for given query directions from the point $\boldsymbol{C}$. Thus, via the GP prior, for a query direction parameter vector $\boldsymbol{x} \in \mathbb{R}^2$, we can obtain a conditional distribution over the distance to the surface along the given direction, and a collection of surface (training) data, $(\boldsymbol{\Psi}, \boldsymbol{d}) = ([\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_M], [d_1, \ldots, d_M])$:

$$q\left(d|\, \boldsymbol{x}, \boldsymbol{C}; (\boldsymbol{\Psi}, \boldsymbol{d})\right) \sim \mathcal{N}\left(\mu_{d|\boldsymbol{x}}, \sigma^2_{d|\boldsymbol{x}}\right), \tag{5}$$

where,

$$\begin{aligned} \mu_{d|\boldsymbol{x}} &= \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{\Psi})\boldsymbol{K}(\boldsymbol{\Psi}, \boldsymbol{\Psi})^{-1}\boldsymbol{d}, \\ \sigma^2_{d|\boldsymbol{x}} &= k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{\Psi})\boldsymbol{K}(\boldsymbol{\Psi}, \boldsymbol{\Psi})^{-1}\boldsymbol{K}(\boldsymbol{\Psi}, \boldsymbol{x}). \end{aligned} \tag{6}$$

To discharge notation, we will henceforth omit the training data from the likelihood expression, $q\left(d|\, \boldsymbol{x}, \boldsymbol{C}; (\boldsymbol{\Psi}, d)\right)$, and simply use, $q\left(d|\, \boldsymbol{x}, \boldsymbol{C}\right)$.

3.3 Surface coverage with mixtures of likelihoods

It becomes evident that for objects with more complex shapes, a single GP model will not suffice, because there would exist rays emanating from the reference point that intercept the exterior surface of the object in multiple points (e.g. Fig. 3). To cope with this issue, we introduce multiple reference points which we distribute preferably in the interior of the object in a way that achieves coverage of the entire surface. Interior reference points ensure that every direction corresponds to a valid training sample. A topology-aware placement of such reference points would require distance-based clustering such as k-means (Lloyd, 1982) or expectation maximization (EM) (Do and Batzoglou, 2008), as it will induce a surface partition that will comprise loosely convex patches that are fully reachable from the respective cluster centers via ray-casting. For each of these reference points, a new GP prior is introduced. Therefore, the complete object surface is modeled by training multiple predictors of local surface patches with reference points placed at the cluster centers.

Formally, the idea is to model the likelihood of a 3D point $\boldsymbol{P}$ belonging to the surface of the object $\boldsymbol{o}$ as a mixture of normal distributions derived from the corresponding GP priors with

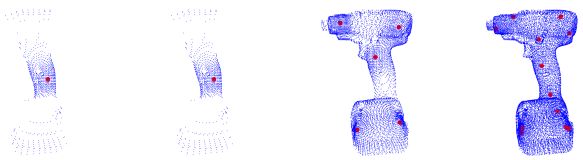
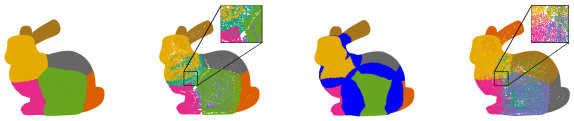


Figure 3. Ray-casting to the objects's surface (blue) from reference points (red) by retaining the first intersection. Denser coverage achieved with increasing number of reference points (left-to-right).



(a) (b) (c) (d)

Figure 4. Assignment of training points to clusters. (a)-(b) Ground truth - reconstructed without overlap. (c)-(d) Ground truth - reconstructed with overlap (blue).

reference points at the cluster centers $\boldsymbol{C}_k$,

$$p(\boldsymbol{P}|\boldsymbol{o}) = \sum_{k=1}^{K} p\left(\boldsymbol{P}|\boldsymbol{C}_k;\boldsymbol{o}\right)\pi(\boldsymbol{C}_k;\boldsymbol{o}), \tag{7}$$

where we choose the likelihood $p\left(\boldsymbol{P}|\boldsymbol{C}_k;\boldsymbol{o}\right)$ to be the GP prior based normal distribution of Eq. (5), i.e.,

$$p\left(\boldsymbol{P}|\boldsymbol{C}_k;\boldsymbol{o}\right) = q\left(\|\boldsymbol{P}-\boldsymbol{C}_k\| \,\middle|\, \boldsymbol{x}_P, \boldsymbol{C}_k\right), \tag{8}$$

and $\boldsymbol{x}_P$ is the direction parameter vector of $\boldsymbol{P}$ with respect to the reference point $\boldsymbol{C}_k$. Furthermore, the weighting probability $\pi(\boldsymbol{C}_k;\boldsymbol{o})$ is modeled as a softmax ratio based on the distance of $\boldsymbol{P}$ from the reference point $\boldsymbol{C}_k$:

$$\pi(\boldsymbol{C}_k;\boldsymbol{o}) = \frac{\exp\left(-\left(\boldsymbol{P}-\boldsymbol{C}_k\right)^T \boldsymbol{Q}_k \left(\boldsymbol{P}-\boldsymbol{C}_k\right)\right)}{\sum_{l=1}^{K}\exp\left(-\left(\boldsymbol{P}-\boldsymbol{C}_l\right)^T \boldsymbol{Q}_l \left(\boldsymbol{P}-\boldsymbol{C}_l\right)\right)}, \tag{9}$$

where $\boldsymbol{Q}_k$ is a covariance measure of the local space around the k-th cluster center (reference point) $\boldsymbol{C}_k$ [1]. In practice, this means that we reconstruct a query point $\boldsymbol{P}$ by choosing the nearest center. This however can give rise to borderline errors in the assignment of query points to reference points. To cope with this problem, we impose inter-cluster overlap in the training (Fig. 4).

3.4 Experimental setup

3.4.1 Datasets We conduct our experiments on two datasets: ShapenetCore (Chang et al., 2015) and IndustryShapes (Sapoutzoglou et al., 2026). ShapeNetCore is a densely annotated subset of the full ShapeNet corpus, containing 51,300 unique 3D models that span across 55 common object categories. We limit our experiments to 3 representative object categories, namely *Planes*, *Chairs* and *Sofas* and use 120 models per category. IndustryShapes is a domain-specific dataset containing a limited set of tools and components that are used in an industrial setting of car-door assembly. For our experiments, we use the *Screwdrivers* object category.

3.4.2 Evaluation metrics To demonstrate the accuracy of our GP mixture-based modeling for surface reconstruction we report the following metrics:

Chamfer distance (d_C) . Measures the similarity between the ground-truth and estimated point sets $\mathbf{P_{gt}}, \mathbf{P_{est}}$, and is defined as:

$$\mathbf{d_C}(P_{gt}, P_{est}) = \frac{1}{|P_{gt}|}\sum_{x\in P_{gt}}\min_{y\in P_{est}}\|x-y\|^2 + \frac{1}{|P_{est}|}\sum_{y\in P_{est}}\min_{x\in P_{gt}}\|x-y\|^2. \tag{10}$$

Precision (*P*). Measures the accuracy of the predicted points by quantifying how closely they match the ground truth. We compute the minimum distance between each predicted point p_{est} and the ground truth set P_{gt}:

$$d_{p_{est}\to P_{gt}} = \min_{p_{gt}\in P_{gt}}\|p_{est}-p_{gt}\|.$$

By defining a distance threshold τ, we can determine the percentage of predicted points whose distance to the ground truth falls within this threshold:

$$\mathbf{P}(\tau) = \frac{|\{p_{est}\in P_{est} : d_{p_{est}\to P_{gt}} < \tau\}|}{|P_{est}|}.$$

In our experiments, we used a threshold of τ = 0.01 as all models are normalized to the unit sphere.

Recall (*P*). Measures the completeness of the reconstruction, quantifying how well the ground truth points are represented in the predicted point cloud. Similar to Precision, we compute the minimum distance between each ground truth point p_{gt} and the predicted set P_{est}:

$$d_{p_{gt}\to P_{est}} = \min_{p_{est}\in P_{est}}\|p_{gt}-p_{est}\|.$$

Using the same distance threshold τ, Recall is the percentage of ground truth points that are within this threshold from the predicted points:

$$\mathbf{R}(\tau) = \frac{|\{p_{gt}\in P_{gt} : d_{p_{gt}\to P_{est}} < \tau\}|}{|P_{gt}|}.$$

We used the same distance threshold of τ = 0.01 for recall as well.

F-score (*F*). The F-score combines Precision and Recall into a single metric, providing a more robust evaluation than either precision or recall alone. The F-score is defined as:

$$\mathbf{F} = \frac{2\mathbf{P}(\tau)\mathbf{R}(\tau)}{\mathbf{P}(\tau)+\mathbf{R}(\tau)}.$$

3.4.3 Data preparation We prepare our input data following the methodology of DeepSDF (Park et al., 2019b) to allow for a fair comparison. In particular, we first normalize the input model to the unit sphere. Subsequently, we sample a dense point cloud from the model's mesh surface by casting rays from a set of virtual cameras placed on a Fibonacci grid (Swinbank and James Purser, 2006) onto the surface of the sphere. Our training and testing input is formed by sub-sampling two completely distinct sparse point clouds containing 10,000 and 30,000 points respectively.

3.4.4 Optimization Our method uses GPyTorch (Gardner et al., 2018) for training and evaluating the GP models. We use

[1] Provided by the EM algorithm, whereas in the case of k-means we use the identity.

	Planes				Chairs				Sofas			
	$d_C \downarrow$	P ↑	R ↑	F ↑	$d_C \downarrow$	P ↑	R ↑	F ↑	$d_C \downarrow$	P ↑	R ↑	F ↑
DeepSDF (Park et al., 2019a)	0.79	90.2	85.5	87.6	0.81	84.2	88.2	86.0	0.57	89.0	**90.0**	89.4
DeepSDF (10k)	0.80	**90.3**	84.5	87.2	0.90	84.5	88.2	86.1	0.56	65.0	60.8	62.5
NKSR (Huang et al., 2023)	0.65	78.4	81.3	79.5	0.41	84.6	**86.3**	85.0	0.45	79.6	82.5	80.8
Ours	**0.14**	87.4	**93.6**	**90.3**	**0.21**	**92.3**	86.5	**89.2**	**0.26**	**92.4**	89.2	**90.7**

Table 1. Comparison with NKSR (Huang et al., 2023) and DeepSDF (Park et al., 2019a) trained with 500K and 10K points per object across shapes of varying complexity for the planes, chairs, and sofas datasets from ShapeNetCore (120 models each). For the NKSR we used the pretrained model on ShapenetCore. We report d_C (Chamfer distance), P (Precision), R (Recall), F (F-score) computed using a sample of 30k points.

	Planes				Chairs				Screwdriver			
	$d_C \downarrow$	P ↑	R ↑	F ↑	$d_C \downarrow$	P ↑	R ↑	F ↑	$d_C \downarrow$	P ↑	R ↑	F ↑
Polynomial (p=3)	1.56	35.3	58.0	43.7	3.35	26.7	40.4	31.9	1.69	32.6	32.8	32.7
Periodic	0.38	70.4	86.4	77.5	0.62	60.8	73.8	66.2	3.47	26.6	26.8	26.7
Linear	4.37	13.5	36.2	19.5	9.09	10.9	31.8	16.0	9.02	10.5	14.6	12.2
RBF	0.31	73.0	88.9	80.1	0.47	60.7	69.2	66.4	0.20	86.0	88.4	87.2
RQ	**0.19**	**82.6**	**93.8**	**87.9**	0.51	**68.4**	**82.9**	**74.5**	**0.12**	**95.1**	**96.5**	**95.8**
Matern	0.26	76.3	91.8	83.3	**0.43**	64.7	79.2	70.8	0.15	92.0	93.6	92.8

Table 2. Ablation study of different kernels and their impact on reconstruction accuracy across various shapes. Chamfer distance is multiplied by 10^3.

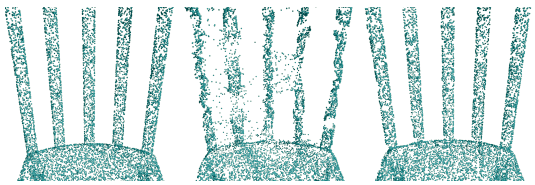
the Adam optimizer (Kinga et al., 2015) with an initial learning rate of 0.1 and employ a learning rate scheduler to dynamically reduce the learning rate by a factor of 0.1 after 10 iterations without improvement in the loss function. All experiments were conducted using a single RTX 4090 GPU with 24GB of VRAM. For an input sparse point cloud of 10,000 points, the GP models typically take about 30 seconds to train on the object, with slight variations depending on the number of reference points. Introducing overlapping regions increases the overall number of training samples.

3.5 Main results

Table 1 presents a quantitative comparison between our proposed shape representation and two state-of-the-art competitors: DeepSDF (Park et al., 2019a) and NKSR (Huang et al., 2023). Our method achieves the lowest Chamfer distance and over 90% F-score across the board. Similarly, competitive Precision and Recall are attained for all object categories, with a remarkable 93.6% Recall for the *Planes* class. As the qualitative comparison of Fig. 6 illustrates, our representation is able to capture the majority of the shape details in a robust manner, while competitors either over-smooth important shape features or miss them completely. It should be noted that we follow an intrinsically different approach. While DeepSDF and NKSR focus on generalization across different objects, the former by learning the latent space of shapes and the latter by combining the conditioning of kernel parameters on the data with kernel ridge regression, our method is more adept in representing details of specific objects. Fig. 7 illustrates the reconstruction results of our representation method applied to models from the IndustryShapes dataset. As demonstrated, our method successfully models complex, thin geometries and preserves challenging topological features, such as holes, to a significant extent.

3.6 Ablation studies

3.6.1 Reference points. We first study the impact that the multitude and the positioning of the reference points might have on the template’s fidelity. To isolate the effect of reference point quantity, Fig. 8 illustrates the Chamfer distance trends for the planes, sofas, and screwdriver classes. For these categories—where automatic distance-based clustering (k-means) was used—increasing the number of reference points generally improves representation quality until reaching a plateau. However, for highly complex topologies with multiple tube-like structures, such as the chairs category, k-means initialization lacks the semantic information needed to efficiently place reference points at strategic locations. Therefore, rather than evaluating the chairs category based on quantity alone, we utilize it to demonstrate the critical importance of ideal reference point placement. This specific limitation highlights the necessity of employing more intelligent algorithms, such as skeletonization, to compute reference centers for complex shapes. As shown in Fig. 5, automatic distance-based placement is inferior to manual placement for such complex objects, though it remains adequate for capturing the underlying topology (see also Sec. 3.7).



(a) (b) (c)

Figure 5. (a) Ground truth, (b) Results obtained from automatically estimating reference point locations (16) with k-means, (c) Results based on manually defined reference points (32).

3.6.2 Kernel choice. To identify a suitable kernel that performs robustly across objects of varying classes and complexities, we conducted structured experiments testing a range of kernel types (Rasmussen, 2004), ranging from a third-degree polynomial to periodic, linear, RBF, Matérn, and Rational Quadratic (RQ). To ensure a fair comparison, all models were evaluated using eight fixed reference points per object. For this ablation, we specifically evaluated five models each from the Planes and Chairs categories alongside two models from the IndustryShapes

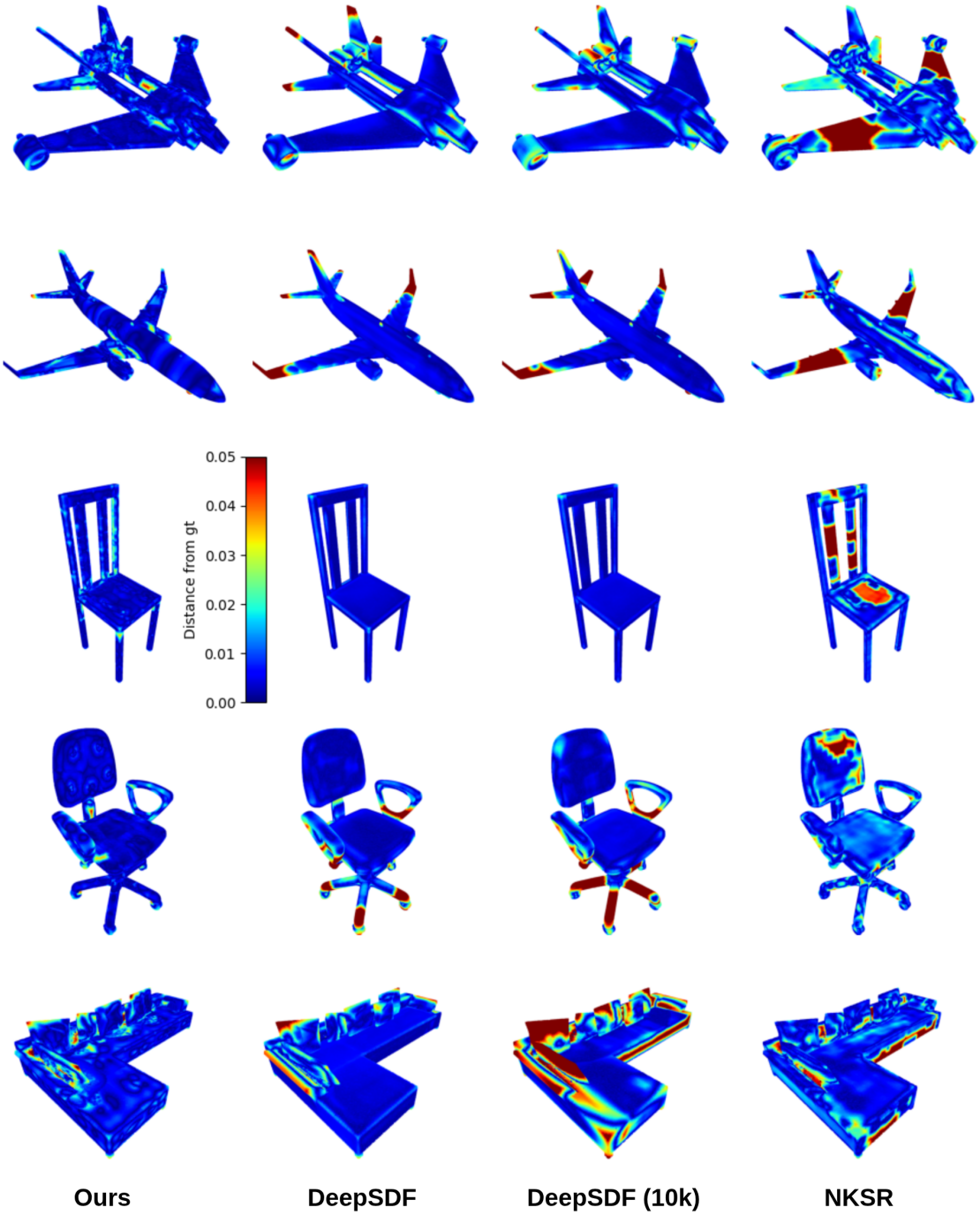


Figure 6. Qualitative comparison of shape representation on the ShapeNetCore dataset. The heatmap quantifies the distance from the ground truth point-cloud to the reconstructed; superimposed into the ground truth point cloud. We demonstrate two examples from the airplanes, chairs and sofas categories with each method. From left to right ours, DeepSDF (Park et al., 2019a), DeepSDF (10k), NKSR (Huang et al., 2023)

Screwdrivers class. These categories were deliberately selected over others (such as Sofas) because they present highly contrasting and challenging geometric features: Planes contain thin, protruding structures, while Chairs possess complex topologies with high-frequency details and tube-like structures. The mean metrics for each category (Chamfer distance, Precision, Recall, and F-score) are detailed in Table 2. The results indicate that while the RBF and Matérn kernels perform well, the RQ kernel exhibits superior overall performance and adaptability across the tested object categories.

3.7 Limitations

Aside the number of points that impact the template fidelity (cf. Sec. 3.6), the locations of the reference points may also influence the quality of the representation. Overall, perturbing the number of reference points and their locations could lead to improvements, particularly when dealing with complex objects. For instance, in the case of the chair depicted in Fig. 5, comparing results obtained from clustering-based estimation of reference points (cf. Sec. 3.1) with those manually defined, a

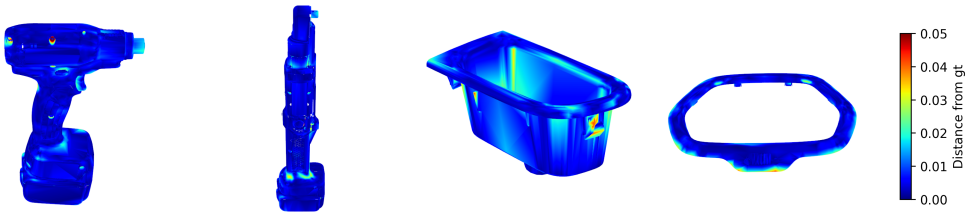


Figure 7. Qualitative results of our method for the models of the IndustryShape dataset.The heatmap quantifies the distance from the ground truth point-cloud to the reconstructed; superimposed into the ground truth point cloud.

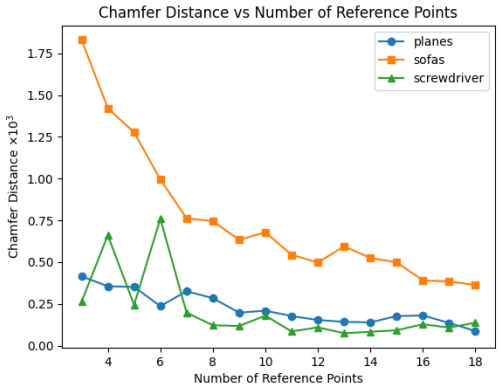


Figure 8. Ablation study for various number of reference points vs shape accuracy based on CD for the planes and sofas object classes of the *ShapeNetCore* dataset and the screwdriver from the IndustryShapes dataset.

more accurate shape representation is derived for the latter.

4. Conclusion & Future work

This work presents a functional, probabilistic shape representation grounded in Gaussian-Process mixtures over directional distance fields. The method provides a lightweight, interpretable, and high-accuracy alternative to neural implicit functions, especially effective under sparse sampling. In future work, we aim at endowing the shape representation algorithm with theoretical guarantees for full coverage of the object surface irrespective of shape complexity via learned mixtures models and fully automating the choice of reference points.

Acknowledgments

This work was partially funded by the European Union’s Horizon Europe programme SOPRANO (GA No 101120990) and PANDORA (GA No 101135775).

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L., 2017. Learning representations and generative models for 3d point clouds.

Aumentado-Armstrong, T., Tsogkas, S., Dickinson, S., Jepson, A., 2022. Representing 3d shapes with probabilistic directed distance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19321–19332.

Bishop, C. M., 2006. *Pattern recognition and machine learning*. 2.

Brock, A., Lim, T., Millar, R., Nicholas, W., 2016. Generative and discriminative voxel modeling with convolutional neural networks. *NeurIPS 3D Deep Learning Workshop*, 1–9.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Cremers, D., 2015. Image Segmentation with Shape Priors: Explicit Versus Implicit Representations. *Handbook of Mathematical Methods in Imaging*, 2, 1909–1944.

Damianou, A., Lawrence, N. D., 2013. Deep gaussian processes. *Artificial intelligence and statistics*, PMLR, 207–215.

Do, C. B., Batzoglou, S., 2008. What is the expectation maximization algorithm? *Nature biotechnology*, 26(8), 897–899.

Dragiev, S., Toussaint, M., Gienger, M., 2011. Gaussian process implicit surfaces for shape estimation and grasping. *2011 IEEE Intl. Conf. on Robotics and Automation*, 2845–2850.

Farshian, A., Götz, M., Cavallaro, G., Debus, C., Nießner, M., Benediktsson, J. A., Streit, A., 2023. Deep-Learning-Based 3-D Surface Reconstruction—A Survey. *Proc. of the IEEE*, 111(11), 1464-1501.

Feng, B. Y., Zhang, Y., Tang, D., Du, R., Varshney, A., 2022. Prif: Primary ray-based implicit function. S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner (eds), *Computer Vision – ECCV 2022*, Springer Nature Switzerland, Cham, 138–155.

Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A., 2022. Plenoxels: Radiance fields without neural networks. *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, 5501–5510.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., Wilson, A. G., 2018. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31.

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., Eslami, S. M. A., 2018. Conditional neural processes. J. Dy, A. Krause (eds), *Proc. of the 35th International Conference on Machine Learning*, Proc. of Machine Learning Research, 80, PMLR, 1704–1713.

Has, S., 2023. Gradient COBRA: A kernel-based consensual aggregation for regression. *arXiv preprint arXiv:2310.01173*.

Huang, J., Gojcic, Z., Atzmon, M., Litany, O., Fidler, S., Williams, F., 2023. Neural kernel surface reconstruction. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4369–4379.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., Harandi, M., 2015. Kernel Methods on Riemannian Manifolds with Gaussian RBF Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(12), 2464-2477.

Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4), 139–1.

Kinga, D., Adam, J. B. et al., 2015. A method for stochastic optimization. *International conference on learning representations (ICLR)*, 5, San Diego, California;, 6.

Liao, Y., Donné, S., Geiger, A., 2018. Deep marching cubes: Learning explicit surface representations. *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2916–2925.

Lloyd, S., 1982. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2), 129–137.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A., 2019. Occupancy networks: Learning 3d reconstruction in function space. *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 4455–4465.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., Ng, R., 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1), 99–106. https://doi.org/10.1145/3503250.

Park, J. J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S., 2019a. Deepsdf: Learning continuous signed distance functions for shape representation. *Proc. of the IEEE/CVF Conf. on computer vision and pattern recognition*, 165–174.

Park, J. J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S., 2019b. Deepsdf: Learning continuous signed distance functions for shape representation. *The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Paschalidou, D., Gool, L. V., Geiger, A., 2020. Learning Unsupervised Hierarchical Part Decomposition of 3D Objects From a Single RGB Image. *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1057-1067.

Pérez-Cruz, F., Van Vaerenbergh, S., Murillo-Fuentes, J. J., Lázaro-Gredilla, M., Santamaria, I., 2013. Gaussian processes for nonlinear signal processing: An overview of recent advances. *IEEE Signal Processing Magazine*, 30(4), 40–50.

Rasmussen, C. E., 2004. *Gaussian Processes in Machine Learning*. Springer Berlin Heidelberg, 63–71.

Rosenfeld, A., Pfaltz, J., 1968. Distance functions on digital pictures. *Pattern Recognition*, 1(1), 33-61. https://www.sciencedirect.com/science/article/pii/0031320368900137.

Sapoutzoglou, P., Vaggelis, O., Zacharia, A., Sartinas, E., Pateraki, M., 2026. IndustryShapes: An RGB-D Benchmark dataset for 6D object pose estimation of industrial assembly components and tools. *arXiv:2602.05555*.

Shawe-Taylor, J., Cristianini, N., 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Swinbank, R., James Purser, R., 2006. Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 132(619), 1769–1793.

Titsias, M., 2009. Variational learning of inducing variables in sparse gaussian processes. D. van Dyk, M. Welling (eds), *Proc. of the Twelfth International Conference on Artificial Intelligence and Statistics*, Proc. of Machine Learning Research, 5, PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 567–574.

Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., Theobalt, C., 2020. Patchnets: Patch-based generalizable deep implicit 3d shape representations. A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (eds), *Computer Vision – ECCV 2020*, Springer International Publishing, Cham, 293–309.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.-G., 2018. Pixel2mesh: Generating 3d mesh models from single rgb images. *Proc. European Conf. Computer Vision – ECCV 2018*, 55–71.

Williams, C. K., Rasmussen, C. E., 2006. *Gaussian processes for machine learning*. 2Number 3, MIT press Cambridge, MA.

Williams, F., Gojcic, Z., Khamis, S., Zorin, D., Bruna, J., Fidler, S., Litany, O., 2021. Neural fields as learnable kernels for 3d reconstruction.

Wilson, A. G., Hu, Z., Salakhutdinov, R., Xing, E. P., 2016. Deep kernel learning. A. Gretton, C. C. Robert (eds), *Proc. of the 19th International Conference on Artificial Intelligence and Statistics*, Proc. of Machine Learning Research, 51, PMLR, Cadiz, Spain, 370–378.

Yenamandra, T., Tewari, A., Yang, N., Bernard, F., Theobalt, C., Cremers, D., 2024. Fire: Fast inverse rendering using directional and signed distance functions. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3065–3075.

Zheng, Z., Yu, T., Dai, Q., Liu, Y., 2021. Deep implicit templates for 3d shape representation. *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 1429–1439.