Task-informed grasping of partially observed objects

Cristiana de Farias, Brahim Tamadazte, Maxime Adjigble, Rustam Stolkin, Naresh Marturi

Task-informed grasping of partially observed objects

Cristiana de Farias, Brahim Tamadazte, Maxime Adjigble, Rustam Stolkin, Naresh Marturi

***Abstract*—In this paper, we address the problem of task-informed grasping in scenarios where only incomplete or partial object information is available. Existing methods, which either focus on task-aware grasping or grasping under partiality, typically require extensive data and long training durations. In contrast, we propose a one-shot task-informed methodology that enables the transfer of grasps computed for a stored object model in the database to another object of the same category that is partially perceived. Our method leverages the reconstructed shapes from Gaussian Process Implicit Surfaces (GPIS) and employs the Functional Maps (FM) framework to transfer task-specific grasping functions. By defining task functions on the objects' manifolds and incorporating an uncertainty metric from GPIS, our approach provides a robust solution for part-specific and task-oriented grasping. Validated through simulations and real-world experiments with a 7-axis collaborative robotic arm, our methodology demonstrates a success rate exceeding 90% in achieving task-informed grasps on a variety of objects.**

***Index Terms*—Grasping, perception for grasping and manipulation, task planning, dexterous manipulation.**

I. INTRODUCTION

In the past decade, extensive research has been dedicated to enhancing robotic manipulation capabilities, targeting rigid and, more recently, deformable objects. Yet, reliably grasping objects, whether known or unknown, within unstructured environments remains a persistent challenge. Many researchers have proposed solutions to this broad grasping problem by leveraging advancements in sensor and computational capabilities [1]–[3]. These solutions, which can be either analytical or data-driven, leverage visual features or even shape uncertainty while aiming at achieving stable grasps on the surface of rigid objects. Despite their success in handling complex objects, many of these methods struggle in challenging conditions, e.g., in the case of non-rigid deformation or partial observation. Furthermore, real-world applications often demand the ability to grasp specific parts of an object. For instance, gripping a knife by its handle facilitates cutting tasks, while handling the blade is suitable for handover tasks.

Attempting to address these challenges, recent studies have explored grasp affordances [4], [5] and part-based grasping [6], [7], alongside task-aware grasping [8]–[10]. These methods range from data-driven techniques to those with limited generalisation capabilities. Closely aligned with our approach, some studies leverage geometric similarities for grasp transfer in environments that are partially observed. [10] investigates bolt repositioning and insertion in cluttered scenes using dense correspondence, while [11] emphasises grasp transfer across different object categories based on common features like lids or handles. Our approach stands apart by transferring high-level task functions to guide a search for feasible grasps on the target object's manifold.

A common challenge in part-specific or task-based grasping is the reduced grasp success rate when the manipulated object diverges from its template or its learned characteristics. This challenge can be addressed by directly formulating grasp hypotheses through exploration of object shapes and transferring these hypotheses across objects [12]–[17]. However, very few studies have explored this approach for non-rigid objects. Particularly, [13]–[15], [18] have all employed frameworks based on *Coherent Point Drift* (CPD) to perform non-rigid shape matching and grasp transfer. In our previous work [19], we demonstrated the effectiveness of the Functional Maps (FM) framework [20] in efficiently transferring grasps from a region of an object to its deformed configurations. FM matches shapes by identifying correspondences between geometric functions on a simplified basis. This choice leads to more straightforward convex least-square optimisation problems and adds greater flexibility when regularising maps. In [19], we also showed that FM-based grasp transfer significantly outperforms state-of-the-art methods employing CPD paradigms. Nevertheless, despite its efficiency, our previous method can only transfer grasps between fully observed shapes.

In this paper, we propose a new approach for task-informed grasping of objects that are only partially observed by a depth camera. Our approach enhances the robot's understanding of the object and enables more informed grasping decisions. It leverages shapes estimated through Gaussian Process Implicit Surfaces (GPIS) and the FM framework to facilitate information transfer between a full model of a reference object stored in a database and a *different* partially observed (potentially deformed) object from the same class. Specifically, we define task functions to obtain the graspable region for a task-informed grasp. For example, when the robot is tasked with pouring using a bottle, we define the grasp region for this task through a function over the bottle's manifold, i.e., the middle of the bottle has higher values, indicating the graspable region. This task function can then be transferred to another partially observed bottle with different shape, allowing the robot to execute a similar grasp on it. Additionally, we use an uncertainty metric from GPIS to improve grasp stability. In summary, our major contributions are:

Manuscript received: March 28, 2024; Revised June 25, 2024; Accepted July 29, 2024. This letter was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the EPSRC under grant EP/P01366X/1, and in part supported by CHIST-ERA under Project EP/S032428/1 PeGRoGAM. *(Corresponding author: Cristiana de Farias.)*

C. de Farias, M. Adjigble, R. Stolkin, and N.Marturi are with the Extreme Robotics Laboratory, School of Metallurgy and Materials, University of Birmingham, Edgbaston, B15 2TT, UK. (email: cristianafar@gmail.com; m.k.j.adjigble@bham.ac.uk; r.stolkin@bham.ac.uk; n.marturi@bham.ac.uk).

B. Tamadazte is with the Sorbonne Université, CNRS UMR 7222, Inserm U1150, ISIR, F-75005, Paris, France. (email: b.tamadazte@cnrs.fr).

This article has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Digital Object Identifier (DOI): see top of this page.

- We propose a method that uses GP in conjunction with FM to transfer grasps between objects of the same class, even with only a partial view of the target object.
- We demonstrate how to incorporate uncertainty into the grasping process to expand the grasping area of a partially observed object while also biasing the grasps towards more stable regions.

Our method is validated through simulation and real-world experiments using a 7-axis robot fitted with a parallel jaw gripper and a wrist-mounted depth camera. Our evaluation assesses how well the method handles a variety of items within the same object class, such as different types of bottles or similarly shaped fruit. The results, summarised in Section IV, demonstrate the effectiveness of our method in transferring task-informed grasps to partially observed objects. Specifically, it showcases a success rate of over $\mathbf{90}\%$ for grasping objects in the correct regions to accomplish a given task.

II. PROBLEM AND BACKGROUND

A. Problem Description

Our primary focus is to transfer grasp between objects belonging to the same category. Specifically, we aim to transfer task-specific grasp execution cost functions from a database of commonly used objects to partially observed similar objects. Let $\mathbf{P}_{db} = \{\mathbf{p}_i^{db} | i = 1, 2, \dots M\}$ denote an M-sized point cloud representing an object in the database, and $\mathcal{S}_{db}$ its corresponding Riemannian manifold. We can then define various task cost functions, $\boldsymbol{f}_i : \mathcal{S}_{db} \rightarrow \mathbb{R}$, depicting the graspable regions of the object. Let $\mathbf{P}_{\text{target}} = \{\mathbf{p}_i^{\text{target}} | i = 1, 2, \dots N\}$ be an N-sized partial point cloud representing a similar (potentially deformed) object, and $\mathcal{S}_{\text{target}}$ its manifold. The problem is then formulated as obtaining $\boldsymbol{g} : \mathcal{S}_{\text{target}} \rightarrow \mathbb{R}$ as

$$\boldsymbol{g} = \mathbf{T}\boldsymbol{f}_i \tag{1}$$

where, $\mathbf{T} \in \mathbb{R}^{N,M}$ is the transformation between the points in the manifolds. Once the function is transferred, a set of grasps $\boldsymbol{\mathcal{G}}(\mathbf{P}_{\text{task}})$ can be obtained. Here, the cloud $\mathbf{P}_{\text{task}}$ comes from its corresponding manifold $\mathcal{S}_{\text{task}} \subset \mathcal{S}_{\text{target}}$ defined as $\mathcal{S}_{\text{task}} \triangleq \boldsymbol{g}(\mathbf{x}) > \varepsilon_{\text{task}}$ with $\mathbf{x} \in \mathcal{S}_{\text{target}}$, and $\varepsilon_{\text{task}}$ a threshold value. To address the problem in (1), obtaining the $\mathbf{T}$ is essential. As stated earlier, the FM framework is employed for this purpose.

B. Background Concepts

1) Shape estimation: The shape reconstruction method employed in this work uses Gaussian Process Regression (GPR) as its main basis [21]. Let, $\boldsymbol{\mathcal{X}} = [\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_n]$ where $\mathbf{x}_i \in \mathbb{R}^d$ represents a d-dimensional n input points and $\boldsymbol{\mathcal{Y}} = [y_1, y_2, \dots, y_n]$, with $y \in \mathbb{R}$, be the n observed output of a system, such that the data pair $\{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$ is related as $y_i = h(\mathbf{x}_i) + \epsilon$. Here $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is the observation's noise (given by a normal distribution with variance σ_ϵ) and $h(\mathbf{x}_i)$ is the real, noiseless output of the system. From $\{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$, the GPR for a new data point $\mathbf{x}^*$ can be defined by a mean function $m(\mathbf{x})$, and covariance function, $k(\mathbf{x}, \mathbf{x}')$. Thus, $f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}') \right)$ is described as

$$\begin{aligned} \bar{h}(\mathbf{x}^*) &= k(\boldsymbol{\mathcal{X}}, \mathbf{x}^*)^T \boldsymbol{\Sigma} \boldsymbol{\mathcal{Y}} \\ \mathbb{V}(\mathbf{x}^*) &= k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{x})^T \boldsymbol{\Sigma} k(\mathbf{x}^*, \mathbf{x})^T \end{aligned} \tag{2}$$

with $\boldsymbol{\Sigma} = \left(k(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}) + \sigma_\epsilon^2 \boldsymbol{I}\right)^{-1}$ [21]. $\bar{h}(\mathbf{x}^*)$ is the predicted mean at $\mathbf{x}^*$ and $\mathbb{V}(\mathbf{x}^*)$ is the predicted variance.

For the shape reconstruction problem, GPR can be employed with implicit surfaces as described in [22]. In this context, the scalar function $p : \mathbb{R}^d \rightarrow \mathbb{R}$ can define a $(d-1)$-dimensional manifold $\mathcal{M}$ whenever the function intersects a particular value, such as zero. Consequently, the data pair $\{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$ is defined by the piece-wise function as follows

$$h_{\text{IS}}(\mathbf{x}) = \begin{cases} -1, & \text{if } \mathbf{x} \text{ is below the surface} \\ 0, & \text{if } \mathbf{x} \text{ is on the surface} \\ 1, & \text{if } \mathbf{x} \text{ is above the surface.} \end{cases} \tag{3}$$

where $\mathbf{x}$ is derived from the object's point cloud along with supplementary synthetic points strategically positioned at the scene's boundaries. This ensures that the GPIS remains both closed and bounded. With $\{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$, the GPIS surface is obtained by $h(\mathbf{x}) \sim \mathcal{GP}\left(0, k_{\text{tp}}(\mathbf{x}, \mathbf{x}') \right)$, where $k_{\text{tp}}(\mathbf{x}, \mathbf{x}') = 2r^3 - 3Rr^2 + R^3$, is the Thin Plate (TP) kernel in which $r = \|\mathbf{x} - \mathbf{x}'\|$ and R is the maximum possible value of r [22]. TP kernel is effective at modeling smooth surfaces, making it suitable for applications like GPIS that require continuous surface reconstruction. Studies have demonstrated its ability to model object shapes accurately with minimal hyperparameter tuning, relying solely on the maximum distance R between points [23]. This aligns well with our application.

2) Functional Map Correspondence: FM aims to find the transformation $\mathbf{T}_p : \mathcal{S}_2 \rightarrow \mathcal{S}_1$ that maps each of the vertices of shape $\mathcal{S}_2$ onto shape $\mathcal{S}_1$. While it might seem intuitive to seek a permutation matrix directly from the vertices of both shapes, such a direct approach is effectively an NP-hard quadratic assignment problem. Alternatively, FM has been successfully adopted in recent years to efficiently solve the problem of non-rigid shape matching between near-isometric shapes [24]. The core principle is that optimising real-valued function correspondences between shapes is more tractable than directly dealing with point correspondences. In this way, FM encodes shapes on a low-rank, compact basis, which allows for the shape matching to be defined as a linear least-squares problem. Mainly, it facilitates the straightforward integration of linear constraints to regulate the mapping.

Given two shapes, $\mathcal{S}_1$ and $\mathcal{S}_2$, below we summarise the FM pipeline to obtain the $\mathbf{T}_p$ point-to-point map between the shapes as well as the FM matrix representation, *i.e.*, $\mathbf{C}$.

i) For each shape, compute a set of orthonormal basis and encode their coefficients as the columns $\boldsymbol{\Phi}_{\mathcal{S}_1}$ and $\boldsymbol{\Phi}_{\mathcal{S}_2}$. Generally, these will be the first n eigenfunctions provided by the Laplace-Beltrami (LB) operator [20]. The LB basis is invariant to isometries and rigid motions. It is also easy to compute on 3D meshes, making it a natural multi-scale way to approximate functions.

ii) Compute a set of descriptor vector functions that are expected to be approximately preserved by the yet-to-be-determined map. In their respective basis, each of these descriptor functions can be denoted as $\mathbf{f}_i \in \boldsymbol{\Phi}_{\mathcal{S}_1}$ and $\mathbf{h}_i \in \boldsymbol{\Phi}_{\mathcal{S}_2}$, and their coefficients can be stored in the matrices $\mathbf{F}$ and $\mathbf{H}$, respectively. A commonly used descriptor function is the wave kernel signature [25].

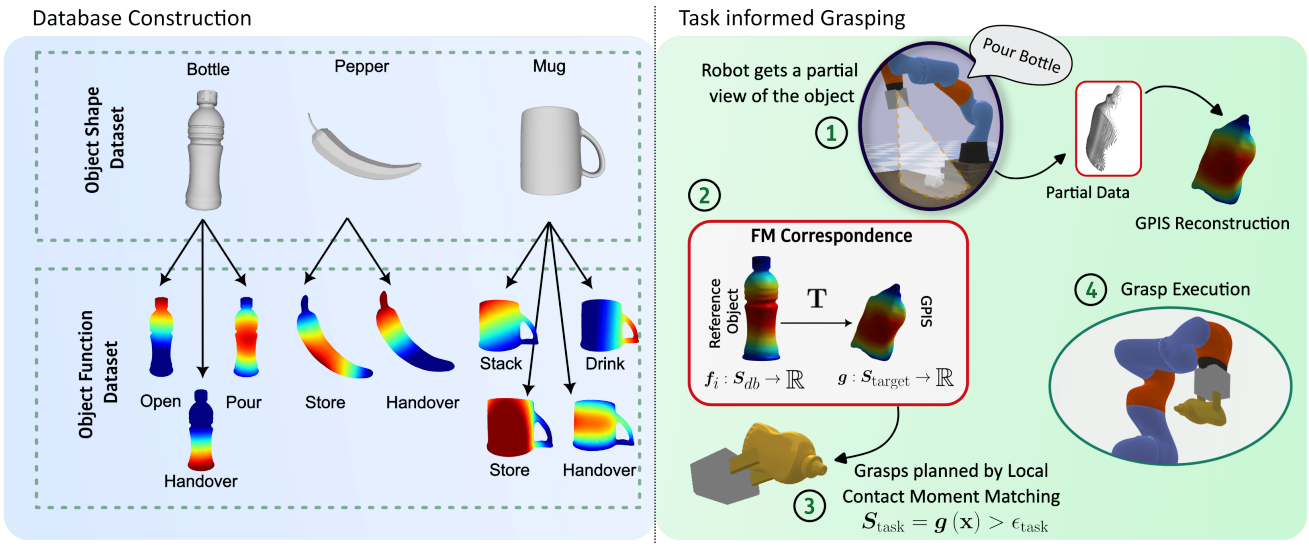

Fig. 1. Pipeline of the proposed task-informed grasping. It consists of a **Database construction** module, where a database of object shapes and task functions are defined and a **Task informed grasping** module, where functions are transferred, and the grasps are executed. A pouring task is shown as an example.

iii) Find the optimal value of $\mathbf{C}$ in the LB-basis by solving

$$\mathbf{C} = \underset{\mathbf{C}}{\operatorname{argmin}}(\alpha_1 E_{\text{DP}}(\mathbf{C}) + \alpha_2 E_{\text{REG}}(\mathbf{C}) \qquad (4)$$

where α_1 and α_2 are scaling terms, $E_{\text{DP}}(\mathbf{C}) = \|\mathbf{CF} - \mathbf{H}\|^2$ is the descriptor preservation energy and E_{REG} is a regularisation term which adds extra constraints to promote better overall structural properties of the map.

iv) Refine the FM and revert it to the point-to-point correspondence vector $\mathbf{T}_p$ by using methods such as the *ZoomOut* iterative upsampling process proposed in [26].

III. METHODOLOGY

A. Proposed Method Pipeline

Our method's pipeline, illustrated in Fig. 1, comprises two modules: (i) **Database construction** (Section III-B) – where a database of selected objects and their associated task functions are defined, and (ii) **Task informed grasping** (Section III-C) – where (1) is solved and a set of k grasps, $\boldsymbol{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_k\}$ is calculated in the high probability region of $\mathbf{P}_{target}$. Various functions indicating the grasp region are created during the database construction step. For instance, the bottle object can have three functions defined for different tasks: one for pouring, another for opening, and a third for handing over the object. In the task-informed phase, a user selects an object and a corresponding task from the database. Later, the robot captures a partial view of the scene object, and its GPIS is constructed. It is assumed that this object belongs to the same class of objects as the reference. Subsequently, the FM transfers $\boldsymbol{f}_i$ (e.g., "pour" function in Fig. 1) to the object's GPIS. Finally, grasps are planned and executed within the preferred region of the current object. Note that the FM framework takes object meshes as input.

This work employs a training-free, direct approach using GPIS for shape estimation prior to shape matching. Notably, GPIS encodes uncertainty in shape reconstruction, aiding in identifying the optimal graspable region of the object.

B. Database Construction Module

During this offline step, we construct the dataset comprising reference objects and their associated task functions. In practice, we generate the manifolds $\mathcal{S}_{db}$ for each object by capturing their point clouds and constructing their meshes using the screened Poisson method [27]. To populate the dataset, a robot with a wrist-mounted camera captures an object's ground plane segmented point cloud by registering clouds from multiple viewpoints [28]. Once an object's point cloud is obtained, the GPR defined in (2) with the Squared Exponential (SE) kernel is used to generate task functions $\mathbf{f_i} \in \mathcal{S}_{db}$. The SE kernel is ideal for this task as it produces smooth, continuous, and infinitely differentiable functions [21]. Here, the data points $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}$ are user-defined. For any given location $\boldsymbol{\mathcal{X}}$ on the object's surface, the corresponding $\boldsymbol{\mathcal{Y}}$ value ranges from 0 to 1. Values near 1 denote regions more suitable for grasping, while values near 0 indicate regions less suitable for grasping. These points are typically chosen based on the user's understanding of the task requirements, ensuring that the selected regions are appropriate for the intended grasp. In Fig. 1 (left side), functions for three different objects are illustrated, with red-colored areas being the graspable zones. Typically, 2 to 5 data points are enough to create these task functions.

C. Task-informed Grasping Module

A prerequisite is that a reference object $\mathcal{S}_{db}$ and a task function $\boldsymbol{f}_i$ are selected from the database. As depicted in Fig. 1, during this phase, the robot partially observes the object by capturing the scene's point cloud from a single viewpoint. It is worth noting that the extent of the object's surface captured in a single view can vary from one object to another. Therefore, we analyse the degree of partiality our method can handle in Section IV-C2. Upon capturing the partial cloud, the corresponding implicit surface h_{IS} is computed using (3). Subsequently, for a grid defined over the workspace, the regression is applied and the manifold

$\mathcal{S}_{\text{GPIS}} \triangleq \{\mathbf{x}|0-\delta \leq \mathbf{x} \leq 0+\delta\}$ is defined, with δ being a small threshold value. Once $\mathcal{S}_{\text{GPIS}}$ is available, the FM process outlined in Section II-B2 is used to obtain the LBO basis $\Phi_{\mathcal{S}_{db}}$ and $\Phi_{\mathcal{S}_{\text{GPIS}}}$, along with the functional map $\mathbf{C}$ using (4).

As mentioned earlier, FM is well-suited for transferring information across various configurations of an object. This transfer can be achieved through the equation

$$\boldsymbol{g} = \Phi_{\mathcal{S}_{\text{GPIS}}} \mathbf{C} \left(\Phi_{\mathcal{S}_{db}}\right)^{\dagger} \boldsymbol{f}_i \tag{5}$$

where, † denotes the *Moore-Penrose* pseudo-inverse. Using (1) and (5), $\mathbf{T}$ can be given as

$$\mathbf{T} = \Phi_{\mathcal{S}_{\text{GPIS}}} \mathbf{C} \left(\Phi_{\mathcal{S}_{db}}\right)^{\dagger}. \tag{6}$$

Next, we obtain the point cloud $\mathbf{P}_{\text{task}}$ from $\mathcal{S}_{\text{task}} \triangleq g(\mathbf{x}) > \varepsilon_{\text{task}}$. Given the transfer of information between $\mathcal{S}_{db}$ and $\mathcal{S}_{\text{GPIS}}$, it follows that $\mathbf{x} \in \mathcal{S}_{\text{GPIS}}$. However, to ensure grasp stability, grasps should be planned over surfaces where object information is available. One way to do that is by planning directly in a subset corresponding to the partial cloud $\mathbf{P}_{\text{target}}$. Thus, the graspable region is the cloud obtained from the intersection between $\mathbf{P}_{\text{task}}$ and $\mathbf{P}_{\text{target}}$. That is

$$\mathbf{P}_{\text{task}}^{\text{partial}} = \mathbf{P}_{\text{task}} \cap \mathbf{P}_{\text{target}}. \tag{7}$$

To obtain the grasp set $\mathcal{G}(\mathbf{P}_{\text{task}}^{\text{partial}})$, we utilise our Local Contact Moment[1] (LoCoMo) based grasp planner presented in [1]. Using zero moments shift, it generates grasp hypotheses by assessing local similarities between an object's surface and gripper fingers. Grasps are then presented in a ranked list based on a shape similarity score. The set of k grasps is denoted as $\boldsymbol{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_k\}$ in which each grasp is $\mathcal{G}_j = \{\mathbf{q}_j, r_j\}$ where $\mathbf{q}_j \in \text{SE}(3)$ is the gripper pose in the world frame and r_j is the LoCoMo ranking score, calculated as in [1]. Once grasp poses are generated, the robot selects the top-ranked and best feasible pose for execution.

D. Grasping in Uncertain Regions

Using (7) to define the object's graspable area ensures stability by limiting grasp planning to regions with confirmed existence. However, it requires that the observed partial region be graspable, *i.e.*, it should have a corresponding region that fits within the gripper's stroke. Planning grasps directly on the GPIS reconstructed surface provides a way to mitigate these limitations. Nevertheless, this strategy might lead to sub-optimal planning, especially in areas characterised by high uncertainty. GPIS-reconstructed surfaces offer a specific advantage in tackling this issue: the uncertainty that is encoded in the estimated shapes. Specifically, from (2), as we only consider new data points $\boldsymbol{x}^*$ falling within $\mathcal{S}_{\text{GPIS}}$, then $\mathbb{V}$ and $\boldsymbol{g}$ are in the same manifold and can be merged in a way that skews the grasp area towards regions where uncertainty is lower while effectively increasing the graspable region. This can be accomplished by calculating a new function:

$$u_{task}(\mathbf{x}) = \left(1 - K_u \tilde{v}(\mathbf{x})\right) g(\mathbf{x}) \tag{8}$$

[1]Code available at: `github.com/maximeadjigble/grasplocomo`

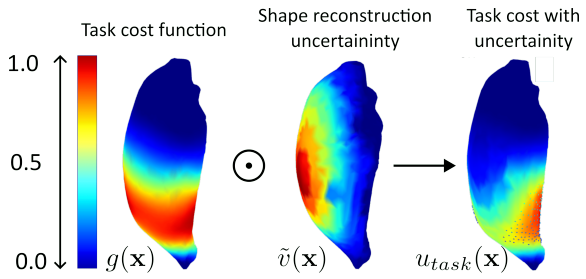

Fig. 2. The individual terms for the task cost function $g(\mathbf{x})$ and the uncertainty function ($\tilde{v}(\mathbf{x})$) are defined on the same manifold. Using equation (8), these terms are combined to form $u_{\text{task}}(\mathbf{x})$, still within the same manifold.

with K_u being a scaling factor between 0 and 1, and $\tilde{v}(\mathbf{x})$ the re-scaled variance, which is computed as

$$\tilde{v}(\mathbf{x}) = \frac{\mathbb{V}(x) - \mathbb{V}_{min}}{\mathbb{V}_{max} - \mathbb{V}_{min}}. \tag{9}$$

Using (8) and (9), the procedure from Section III-C for obtaining grasps can be reiterated. The only variation is in using $\mathcal{S}_{\text{task}} \triangleq u_{task}(\mathbf{x}) > \varepsilon_{\text{task}}$ to derive $\mathbf{S}_{\text{task}}$ and $\mathcal{G}(\mathbf{S}_{\text{task}})$. Fig. 2 depicts the task cost functions over reconstructed objects after (8) is applied.

IV. EXPERIMENTAL VALIDATIONS

To evaluate the proposed approach, we first perform an exhaustive analysis of various methodological aspects through simulation. Following this, we demonstrate the proposed grasp transfer technique using a real-robot setup.

A. Setup Description

1) Simulation setup: Our simulation setup built in PyBullet involved a 7-axis robot with a 2-finger gripper, replicating the kinematics of the KUKA iiwa robot and the Schunk PG-70 gripper. We simulated a 3D camera fixed to the robot's end-effector to capture scene data. The robot was moved to ten positions to build reference objects. The point clouds captured at each location were processed and stitched to generate full object meshes. In the case of grasping experiments, the objects are treated as unknown and only partially observable. Hence, we obtained or merged single or dual views based on the visible object region. Note that the operator enables dual view capture before starting the experiments, and the number of views is fixed for a single object. We used PyMeshLab's screened Poisson algorithm for mesh construction from point clouds. GPIS was implemented with the GPy library [29], and for FM, we utilised code from [26]. All experiments were run on an Ubuntu 20.04 machine with an i7 4-core CPU and 32 GB RAM. Currently, it takes ≈ 25 sec for a single run, *i.e.*, to transfer a task function and find a suitable grasp.

We conducted three sets of scenarios to validate our approach and better understand how various elements within our method influence its success. Firstly, we investigated the impact of the uncertainty gain K_u as described in (8) on the final grasping success rate (refer to Section IV-C1). Following, we analysed the influence of the amount of partial data on our method's success (as detailed in Section IV-C2). Lastly, we

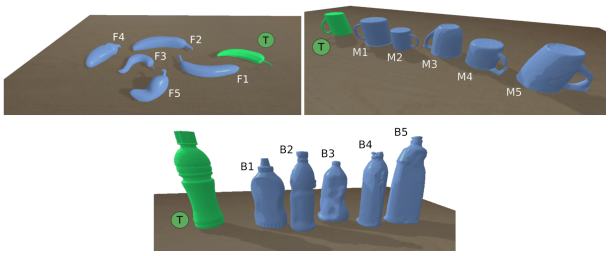

Fig. 3. Dataset of simulation objects. The reference object is shown in green.

Fig. 4. Objects used for real-world trials.

examined the overall grasp success for three different classes of objects: bottles, fruit (e.g., pepper, banana), and mugs. Six different objects were chosen for each class, designating one as the reference. Experiments involved transferring a task function from the reference object to the remaining five objects within the same class. Fig. 3 shows the dataset of objects used for simulation experiments.

2) Real-world setup: Similar to the simulation setup, our real robot setup consists of a 7-axis KUKA iiwa robot equipped with the Schunk PG 70 gripper. A calibrated Ensenso N35 3D camera was affixed to the robot's wrist. Real-world experiments involved either transferring functions between different physical objects of the same class or transferring between the CAD model of template object, *i.e.*, the ones from Fig. 3 and a real object of the same class shown in Fig. 4.

B. Metrics

To analyse grasp success after task-function transfer, we followed the protocol outlined in [30]. It includes three consecutive tests: lifting the object, shaking it by rotating the robot's fifth joint, and rotating it around the robot's seventh joint. The success score presented in the results reflects the combined performance across all three tests, along with verification to ensure the grasp fell within the correct region. Experiments with each test object were repeated five times. Due to variability observed in the sparse GPIS reconstruction, a new point cloud was acquired, and the entire pipeline was executed for each iteration. The results presented in this paper represent the average of all runs. Additionally, we computed the chamfer distance between the full object point cloud and the reconstructed models (in simulation only) to assess the shape estimation. To analyse how partiality affects function transfer, we also performed comparative experiments using the Mean Absolute Error (MAE), which is useful for quantifying the average deviation of the transferred function from the ground truth, thereby providing a measure of accuracy.

C. Simulation Experiments

1) Effect of uncertainty function: To analyse the impact of parameters on our method's success, we examined the variations in the number of grasps and success rates as K_u changed. Values ranging from $0-1$ were used, and experiments were conducted using the bottle `B1` (see Fig. 3) with the functions `Open` and `Pour`.From the plots shown in Fig. 5, it becomes evident that leveraging uncertainty is crucial for the successful transfer of grasp functions.Shape

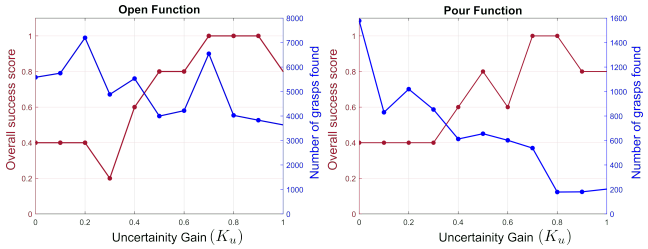

Fig. 5. Impact of gain K_u from (8) on average grasp success rate and (red) and average number of grasps found (blue) for object `B1`.

estimation may interpolate occluded parts into non-existent areas, often associated with high uncertainty. By leveraging the uncertainty information from GPIS reconstruction, we can prioritise grasping areas with more reliable data. However, this method has its limitations. Moving away from parts of the object that have been estimated reduces the number of feasible grasps, potentially leaving no viable options. This reduction is particularly noticeable when $K_u = 1$. Thus, we find that the optimal values for K_u are between 0.7 and 0.9.

2) Analysis of data partiality: In this section, we explore the impact of partial observation, *i.e.*, the extent of the object region captured, on the success of grasp transfer. We used bottle `B1` for this analysis, making three cuts along each of its axes, as illustrated in Fig. 6. The assessment was performed incrementally, beginning with the segment from the first cut, then extending to include the second cut, progressing to the third, and finally considering the entire bottle. It is worth noting that the percentage of points in each section is contingent upon the object's symmetry and the axis along which the cuts are made. The functions `Open` and `Pour` were examined in this context. We also compared the performance of our method with two state-of-the-art correspondence mapping approaches, ICP [31] and CPD [13], by evaluating the MAE obtained with our method against versions of our pipeline where FM is replaced by ICP and CPD. For analysis, we created ground truth functions $\boldsymbol{g}^*$ of both `Open` and `Pour` task functions based on full-to-full transfer between objects. We then computed the MAE between $\boldsymbol{g}^*$ and the transferred data $\boldsymbol{g}$ as $\left(\frac{\boldsymbol{g}^*-\boldsymbol{g}}{N}\right)$, where N is the number of points in the target cloud. Table I summarises the results. Note that, in the table, the mean MAE of both task functions combined is shown.

The results indicate that our FM-based method achieved the highest success and lowest errors across the object's surface. MAE decreased as we approached the full object shape, which

TABLE I
ANALYSING THE IMPACT OF DATA PARTIALITY ON METHOD'S SUCCESS.

Cuts	%Pts.[1]	CD[2]	Part[3] [ICP, CPD, Ours]	MAE [ICP, CPD, Ours]
X-Cuts				
Cut–1	16.99	2.10	[25.0, **40.0**, 30.0]	[0.35, 0.31, **0.24**]
Cut–2	40.57	3.10	[50.0, 45.0, **90.0**]	[0.34, 0.33, **0.09**]
Cut–3	71.38	1.93	[50.0, 50.0, **90.0**]	[0.34, 0.35, **0.08**]
Y-Cuts				
Cut–1	41.42	3.58	[45.0, 40.0, **55.0**]	[0.32, 0.30, **0.18**]
Cut–2	71.58	3.53	[50.0, 50.0, **80.0**]	[0.32, 0.32, **0.09**]
Cut–3	87.73	2.06	[50.0, 50.0, **95.0**]	[0.33, 0.32, **0.08**]
Z-Cuts				
Cut–1	30.22	17.22	[00.0, 00.0, **0.05**]	[**0.48**, **0.48**, **0.48**]
Cut–2	60.11	6.67	[50.0, 25.0, **60.0**]	[0.36, 0.40, **0.27**]
Cut–3	80.33	3.01	[50.0, 50.0, **60.0**]	[0.33, 0.36, **0.30**]

[1] Percentage of points compared to full cloud
[2] Chamfer distance in millimetres between the full and reconstructed models.
[3] Average percentage of success in grasping the correct object part.
* Bold values indicate better performance.

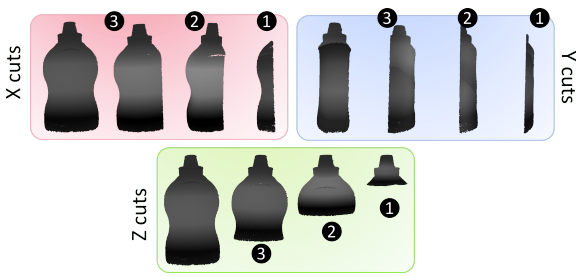

Fig. 6. Cuts made along the X, Y, and Z axes of bottle B1 shown in Fig. 3. Numbers in circles denote cut numbers.

is expected for the FM-based algorithm that accounts for geodesic distances. We also observed that when the position of the target bottle is less aligned with the reference bottle pose, the MAE increases for both ICP and CPD. However, our method is unaffected by this due to the pose-invariant nature of FM. Notably, better performance has been observed for cuts along the X and Y axes than those along the Z-axis. This is due to two main reasons: first, between cut-1 and cut-2, the region corresponding to the `Pour` function is outside the cut, making it harder to find nearby stable grasps. Second, removing larger sections along the Z-axis results in smaller shapes with geometries that deviate further from the template, hindering FM-based matching.

3) Grasp success evaluation: Three classes of objects, each comprising six objects, were used for these tests. Individually, the objects are randomly positioned within the robot's task space, making a partial observation possible with the camera. Five trials were performed with each object for every transferred function. Experiment highlights are presented below, and Table II summarises full results.

- **Mug**: Two poses were selected for all functions. In total, 40 different tests were run, each repeated five times for a total of 200 runs, resulting in $\mathbf{92.8}\%$ success in grasping the correct part and an $\mathbf{85.9}\%$ overall grasp success rate.
- **Bottles**: Three poses were selected for `Open` and `Pour` functions (except when the bottle was symmetric), and one pose for the `Handover` function. In total, 31 experiments were conducted, each repeated five times for a total of 155 runs, resulting in a $\mathbf{96.1}\%$ success rate in grasping the correct part and a $\mathbf{88.8}\%$ overall grasp success.
- **Fruit**: Two poses were selected for all functions. In total, 20 different tests were conducted, each repeated five times for a total of 100 runs, resulting in a $\mathbf{90}\%$ success in grasping the correct part and an $\mathbf{86.8}\%$ overall success.

Fig. 7 illustrates grasping tests for two objects/functions from each class, as well as the transfer from the initial task function to the new object shape.

TABLE II
PERFORMANCE ANALYSIS OF THE PROPOSED APPROACH IN SIMULATION.

Function	C.D.	Part	Lift‡	Rot.‡	Shake‡
Bottle					
Pour	4.41	97.3	93.8	93.8	92.4
Open	3.70	91.1	93.8	92.4	89.8
Handover	3.60	100.0	96.0	96.0	84.0
Total	**3.82**	**96.1**	**94.45**	**94.1**	**88.8**
Mug					
Drink	2.79	100.0	96.0	93.75	85.75
Store	2.83	98.0	94.0	94.0	88.0
Stack	2.70	84.0	92.0	88.0	82.0
Handover	2.96	89.5	96.0	94.0	88.0
Total	**2.82**	**92.8**	**94.5**	**92.4**	**85.9**
Fruit					
Handover	5.49	86.0	92.0	90.0	87.5
Stack	4.08	94.0	92.0	92.0	86.0
Total	**5.51**	**90.0**	**92.0**	**91.0**	**86.8**

‡ Average % of success – reported out of 5 trials.

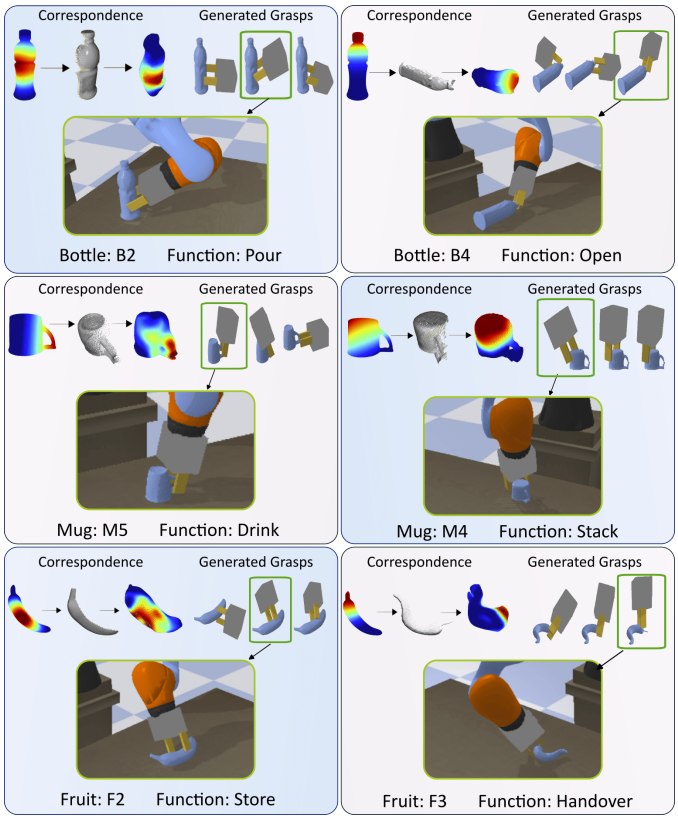

Fig. 7. Six different experiments in simulation showing the transfer, grasp generation (selected grasps are highlighted), grasp execution and stability tests.

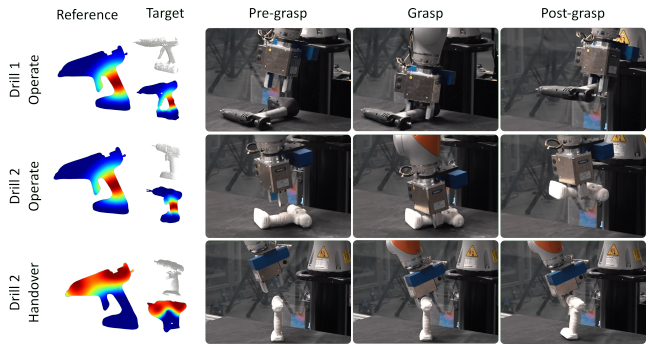

Fig. 8. Function transfer between two real-world objects, showing the transfer from Drill 1 to a partial version of itself and to a completely different drill.

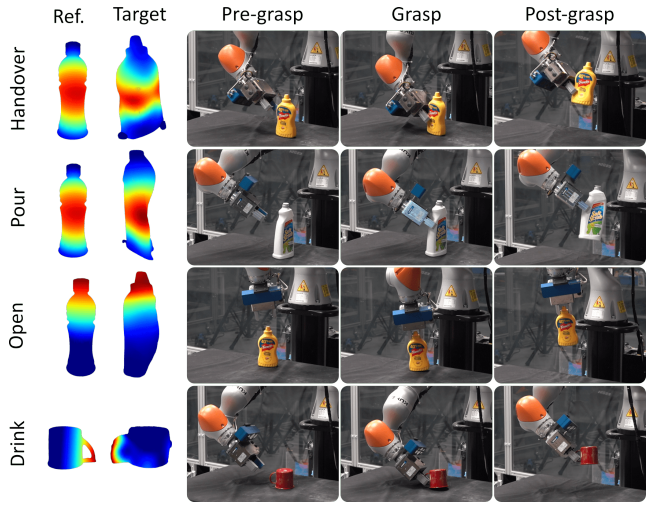

Fig. 9. Results for transferring various task functions to real-world objects. More results are shown in the supplementary video.

D. Real Robot Experiments

The proposed algorithm was tested in two scenarios to validate our approach in a real-world setting. As shown in Fig. 8, functions were transferred between the CAD models of objects (green templates shown in Fig. 3) to real-world objects of the same class shown in Fig. 4. Furthermore, we evaluated the transfer of grasps between two real-world objects as shown in Fig. 9. We utilised the model derived from the complete point cloud of one real object to define a function, which was then transferred to another, similar real-world object. Two additional experiments were conducted to evaluate further the method's applicability and its tolerance for deviation from the reference template. The first experiment, depicted in Fig. 10, involved selecting objects from a similar category but with significant shape differences from the reference. Specifically, as depicted in the figure's top row, we examined the transfer of the `Drink` function from a "cup" (reference object) to a "pitcher" (target object). Although the functions were successfully transferred to the appropriate regions, the outcomes noted some discrepancies. The application of geodesic distances in FM, used for both computing the Laplacian Basis and descriptors on the object's surface, resulted in performance degradation when there were considerable size variations. Fundamentally, the method's efficiency diminishes as the object deviates from being isometric. Addressing this challenge is a focal point of our future work, wherein we aim to explore approaches like normalised scaling or the utilisation of scale-invariant descriptors. In the second experiment, illustrated in the bottom row of Fig. 10, we transferred the `Open` function from a reference bottle to another bottle with a completely different shape, yet within the same scale. The outcome of this experiment was favorable because the geometry of the target bottle can be seen as a near-isometric transformation of the reference, where the isometric distances are largely preserved.

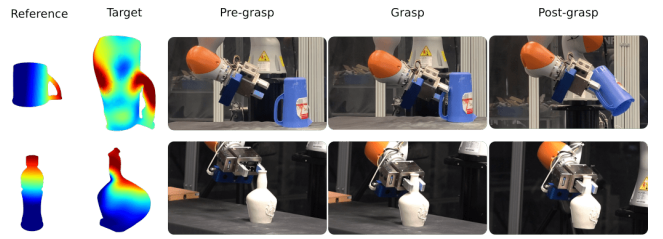

Fig. 10. Function transfer between objects of varying sizes (top row) and shapes (bottom row) within the same class.

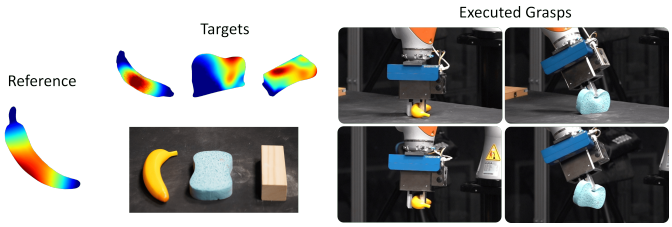

Fig. 11. Analysis of the impact of shape variation on transfer success. `Stack` function from the "Pepper" was transferred to three different objects.

Finally, to address how different the objects can be while still allowing a stable grasp function transfer, we transferred the `Store` function from "pepper" object to "Banana", "Sponge", and "Wooden block", which differ in shape but are similarly scaled. Obtained results are shown in Fig. 11. As in simulation experiments, the transfer to the banana was successful. Similarly, the sponge yielded successful results. However, the transfer to the block was unsuccessful, with no grasps found. Although similar in scale, the block's sharp edges likely interfered with the algorithm's ability to calculate matching features. This issue did not arise with the sponge, which more closely resembles the other shapes.

V. DISCUSSION

In this study, we have demonstrated a method for transferring meaningful functions depicting the grasping region associated with a task between different objects. As shown in Table II, stable grasps for the intended function can be achieved across various objects. Furthermore, as detailed in Section IV-D, this method can be readily adapted to real-world scenarios. Particularly, as the proposed pipeline is based on the FM framework, it ensures pose invariance, facilitating transfers between generic CAD models and real objects (as shown in Fig. 8). Notably, this approach does not require a learning

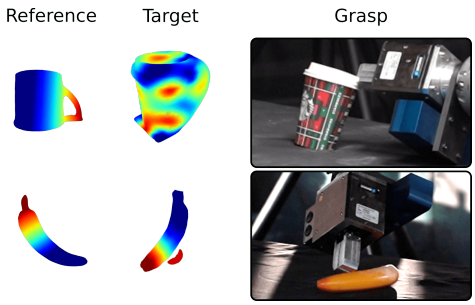

Fig. 12. Failure Cases. (Top) Transfer between the “cup" and a take-away coffee cup. (Bottom) Transfer between “Pepper" and “Banana". An artifact in the reconstruction is visible in the middle image where the grasps are planned.

stage, and satisfactory results can be obtained, provided a reference template is available. Furthermore, we conducted an analysis of the effects of object partiality and the measurement of uncertainty. Although partiality is highly dependent on how the object is occluded rather than the number of points, we have found a correlation between uncertainty and stability.

However, our current method does have some limitations. First, the results' accuracy relies on the reconstruction's quality. If too many artifacts occur during the GPIS or the mesh construction, the results will deteriorate. Some of the common fail-cases are shown in Fig. 12. On the top, the transfer between the “Cup” template and a take-away coffee cup failed due to a mismatch between the reference and reconstructed shapes. On the bottom, despite a successful transfer from “Pepper" to “Banana”, an artifact in the reconstruction led to grasp calculations on a non-existent part of the object, even when incorporating uncertainty into the analysis. The second limitation is that the employed FM pipeline is not scale invariant. As shown in Fig. 10, while certain results are achievable with scale variations, the results are unreliable for objects of different sizes but are of the same class. We plan to address this issue in our future work. Finally, challenges related to symmetry have been noted, particularly with objects like the “Mug" object. The `Stack` function assumed that the grasping area would be on the top. However, as the mug was placed upside-down and the opening was not visible, it failed to differentiate between the two symmetric sides. In future work, we will explore self-symmetric solution spaces to overcome this issue. Despite these limitations, our experiments' high success rate highlights our method's potential to effectively transfer grasp regions across diverse scenarios and objects.

REFERENCES

[1] M. Adjigble, N. Marturi, V. Ortenzi *et al.*, “Model-free and learning-free grasping by local contact moment matching,” in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2018, pp. 2933–2940.

[2] M. Adjigble, C. de Farias, R. Stolkin *et al.*, “Spectgrasp: Robotic grasping by spectral correlation,” in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2021, pp. 3987–3994.

[3] M. Li, K. Hang, D. Kragic, and A. Billard, “Dexterous grasping under shape uncertainty,” *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.

[4] M. Hjelm, R. Detry, C. H. Ek *et al.*, “Representations for cross-task, cross-object grasp transfer,” in *IEEE Int. Conf. Robot. Autom.*, 2014, pp. 5699–5704.

[5] D. Song, C. H. Ek, K. Huebner *et al.*, “Task-based robot grasp planning using probabilistic inference,” *IEEE Trans. Robot.*, vol. 31, no. 3, 2015.

[6] R. Monica and J. Aleotti, “Point cloud projective analysis for part-based grasp planning,” *IEEE Rob. and Auto. Lett.*, vol. 5, no. 3, pp. 4695–4702, 2020.

[7] J. Song, A. Tanwani, J. Ichnowski *et al.*, “Robust task-based grasping as a service,” in *IEEE Int. Conf. on Auto. Sci. Eng.*, 2020, pp. 22–28.

[8] Z. He, N. Chavan-Dafle, J. Huh, S. Song *et al.*, “Pick2place: Task-aware 6dof grasp estimation via object-centric perspective affordance,” in *IEEE Int. Conf. Robot. Auto*. IEEE, 2023, pp. 7996–8002.

[9] D. Song, C. H. Ek, K. Huebner, and D. Kragic, “Task-based robot grasp planning using probabilistic inference,” *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 546–561, 2015.

[10] B. Wen, W. Lian, K. Bekris, and S. Schaal, “Catgrasp: Learning category-level task-relevant grasping in clutter from simulation,” in *2022 IEEE Int. Conf. on Rob. and Auto.*, 2022, pp. 6401–6408.

[11] A. Tekden, M. P. Deisenroth, and Y. Bekiroglu, “Grasp transfer based on self-aligning implicit representations of local surfaces,” *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6315–6322, 2023.

[12] T. Stouraitis, U. Hillenbrand, and M. A. Roa, “Functional power grasps transferred through warping and replanning,” in *IEEE Int. Conf. on Rob. and Auto.*, 2015, pp. 4933–4940.

[13] D. Rodriguez and S. Behnke, “Transferring category-based functional grasping skills by latent space non-rigid registration,” *IEEE Rob. and Auto. Lett.*, vol. 3, no. 3, pp. 2662–2669, 2018.

[14] D. Rodriguez, C. Cogswell, S. Koo *et al.*, “Transferring grasping skills to novel instances by latent space non-rigid registration,” in *IEEE Int. Conf. on Rob. and Auto.*, 2018.

[15] H.-C. Lin, T. Tang, Y. Fan, and M. Tomizuka, “A framework for robot grasp transferring with non-rigid transformation,” in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2018, pp. 2941–2948.

[16] H. Tian, C. Wang, D. Manocha *et al.*, “Transferring grasp configurations using active learning and local replanning,” in *IEEE Int. Conf. on Rob. and Auto.*, 2019, pp. 1622–1628.

[17] C. de Farias, N. Marturi, R. Stolkin *et al.*, “Simultaneous tactile exploration and grasp refinement for unknown objects,” *IEEE Rob. and Auto. Lett.*, vol. 6, no. 2, pp. 3349–3356, 2021.

[18] J. Stückler and S. Behnke, “Efficient deformable registration of multi-resolution surfel maps for object manipulation skill transfer,” in *IEEE Int. Conf. on Rob. and Auto.*, 2014, pp. 994–1001.

[19] C. De Farias, B. Tamadazte, R. Stolkin, and N. Marturi, “Grasp transfer for deformable objects by functional map correspondence,” in *IEEE Int. Conf. on Rob. and Auto.*, 2022, pp. 735–741.

[20] M. Ovsjanikov, M. Ben-Chen, J. Solomon *et al.*, “Functional maps: A flexible representation of maps between shapes,” *ACM Trans. Graph.*, vol. 31, no. 4, 2012.

[21] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[22] O. Williams and A. Fitzgibbon, “Gaussian process implicit surfaces,” in *Gaussian Processes in Practice*, 2006.

[23] G. Z. Gandler, C. H. Ek, M. Björkman *et al.*, “Object shape estimation and modeling, based on sparse gaussian process implicit surfaces, combining visual data and tactile exploration,” *Rob. and Auto. Sys.*, vol. 126, p. 103433, 2020.

[24] M. Ovsjanikov, E. Corman, M. Bronstein *et al.*, “Computing and processing correspondences with functional maps,” in *ACM SIGGRAPH*, 2017.

[25] M. Aubry, U. Schlickewei, and D. Cremers, “The wave kernel signature: A quantum mechanical approach to shape analysis,” in *IEEE Int. Conf. on Comp. Vis. Workshops*, 2011, pp. 1626–1633.

[26] S. Melzi, J. Ren, E. Rodola *et al.*, “Zoomout: Spectral upsampling for efficient shape correspondence,” *arXiv preprint arXiv:1904.07865*, 2019.

[27] M. Kazhdan and H. Hoppe, “Screened poisson surface reconstruction,” *ACM Trans. on Graph.*, vol. 32, no. 3, pp. 1–13, 2013.

[28] N. Marturi, M. Kopicki, A. Rastegarpanah *et al.*, “Dynamic grasp and trajectory planning for moving objects,” *Auto. Rob.*, vol. 43, no. 5, pp. 1241–1256, 2019.

[29] GPy, “GPy: A gaussian process framework in python,” http://github.com/SheffieldML/GPy, since 2012.

[30] Y. Bekiroglu, N. Marturi, M. A. Roa, K. J. M. Adjigble *et al.*, “Benchmarking protocol for grasp planning algorithms,” *IEEE Rob. and Auto. Lett.*, vol. 5, no. 2, pp. 315–322, 2020.

[31] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Trans. on Patt. Anal. and Mach. Intel.*, vol. 14, no. 2, pp. 239–256, 1992.