2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Robotic System Architecture Design for Manipulation of 3D Deformable Objects

Félix Nadon†, Angel J. Valencia†, and Pierre Payeur

School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, ON, Canada
{fnado079, avale073, ppayeur}@uottawa.ca

***Abstract*— This paper presents a system architecture for the robotic manipulation and reshaping of 3D deformable objects. The inherent complexity of manipulating objects with evolving shapes requires the construction and efficient integration of components that are reusable and remain functional under operational scenarios that involve high variability. Such original components are developed and integrated in this paper, notably including a mapping subsystem to aggregate and analyse sensor data for extracting general manipulation and sensing heuristics, as well as a deformation modelling component to predict the effect of candidate robot actions on the shape of an object of interest. A simulation interface is further implemented on top of robotic simulators, allowing a modular creation of environments to evaluate the performance of the aforementioned components and to generate synthetic datasets of 3D deformable object manipulation tasks. These general-use components are integrated with application-specific task and motion planning components in real and simulated environments.**

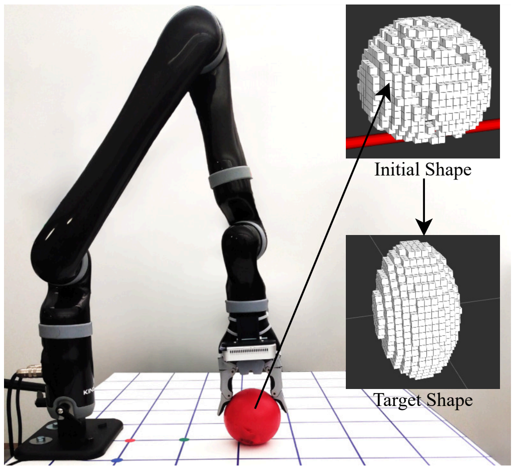

Fig. 1. Jaco manipulator with RealSense camera and deformable object initial and target shapes.

I. Introduction

Robotic manipulation of deformable objects involves the intrinsic challenge that the precise shape of the object under manipulation must be acquired, tracked, predicted, and updated throughout the manipulation process. The complexity of the representation, tracking and prediction of the evolving object shape scales with the dimensionality of the object and may result in prohibitive computational costs for autonomous robot hardware. Volumetric (3D) objects present the additional challenge that their shape cannot be completely captured from a single viewpoint, requiring the merging and interpretation of multiple sensor frames into a coherent representation.

These requirements related to deformable object manipulation add a supplementary layer to the complex interplay of components already present in robotics systems. Notably, it increases the coupling between sensing and acting, as the shape of the manipulated object must be tracked and predicted for each prospective robot action. Contrarily to manipulating rigid objects, the evolving shape of deformable objects becomes part of the world state that must be considered when planning, leading to stronger interdependency between mapping, action planning and motion planning.

The main problem considered in this paper and for the experimental evaluation of the proposed integrated system is a general deformable object reshaping task applicable to both in-hand and large-scale manipulation scenarios. Working in an environment that is initially unknown and partially observable, a robotic system equipped with some configuration of grippers and mobile sensors such as RGB-D cameras must change the shape and pose of a deformable object of interest with unknown material properties to match a user-defined target shape and pose (Fig. 1).

To achieve this, the system must plan sensor movements to build a map of the environment, extract the initial shape and pose of the object of interest, compare it with the target shape and pose, and plan manipulation actions to reshape the object. Multi-step manipulation planning and precise reshaping further require the ability to predict the effect of potential manipulation actions on the shape of the object, as well as continuous tracking of the object shape during manipulation to update the predictions with the true behaviour of the object. Without loss of generality, tactile sensors or force probes may be added to provide additional information in the map of the environment and inform the planning and prediction components.

†Authors contributed equally

Designing a system that addresses these issues while retaining enough modularity and generality to remain reusable between different tasks and applications is, in itself, a significant challenge. In this paper, we introduce a robotic system architecture for the manipulation and controlled reshaping of 3D deformable objects, with a focus on building the perception, mapping, and shape prediction subsystems in a way that is modular and reusable, as well as robot- and task-agnostic.

Section II presents a brief discussion of the relevant literature on manipulation of deformable objects, shape and dynamics modelling, and planning requirements. Section III introduces the proposed architecture from a general perspective before discussing the mapping (III-A) and deformation modelling (III-B) components. The robotic environment and related challenges are discussed in section IV, both for the physical (IV-A) and simulation (IV-B) contexts. Section IV-C presents a sample of results, and section V concludes the paper.

II. Related Works

Robotic manipulation of deformable objects has been studied previously in the reviews of [1]–[3]. While the topic has seen a significant increase in popularity since the publication of these reviews, their general conclusions still hold when looking at more recent works. In particular, a vast majority of works focus on accomplishing a specific task with specific objects and environmental conditions. Different task categories—generally defined by object type and goal such as cable assembly, cloth folding or robotic surgery—are evolving in parallel and there has been little progress toward a global, generalizable system that can tackle the challenges of deformable object manipulation in any scenario. For instance, Zurn et al. [4] recognize the need for a general, modular system, but their design is limited to the 2D shape control of a deformable linear object (e.g., cable) in a fully-observable environment.

Widely available mapping systems, such as OctoMap [5] or the Spatio-Temporal Voxel Layer [6], are typically focused on measuring the occupancy state of large portions of an unknown environment for the purposes of navigating with an autonomous ground or aerial vehicle. While they may also be used for obstacle avoidance in motion planning of robotic manipulators, they are mostly limited to storing occupancy data and do not support functionalities to perform operations such as segmentation, shape comparison, and other computations on the aggregated maps. Thus, they are ill-suited to tasks where an initially unknown and continuously changing deformable object must be tracked and compared to a target shape.

Simulation Application Programming Interfaces (APIs) that support the modular creation of simulation environments are helpful tools to provide repeatable benchmarks for assessing the accuracy and effectiveness of proposed algorithms and systems. Few such APIs exist that support deformable objects, and each is developed for a specific simulator which may not support all required features for the robotic manipulation of 3D deformable objects. An example is DEDO [7], which is developed for PyBullet, but focuses primarily on developing scenes for 2D cloth-like objects. Recently, Orbit (now IsaacLab) [8], built on top of IsaacSim, offers scenes made for 3D deformable objects. However, it still requires a certain level of understanding of the simulator for its usage, given its motivation to serve as a general-purpose scene creation setup.

In our previous work [9], we identified several challenges in sensing, modelling, predicting and controlling the shape of 3D deformable objects and proposed an initial framework for coordinating generic 3D deformable object manipulation tasks. In this paper, we augment this basic framework with original mapping and deformation modelling subsystems, which we integrate with motion planning in a robotic environment.

III. 3D Deformable Object Manipulation System

The proposed system architecture is based on the interactions between five key components shown in Fig. 2. A custom mapping subsystem (section III-A) is developed to address the specific requirements of 3D deformable object manipulation by collecting and analysing sensor data to extract the current state of the object of interest, the obstacles present in the environment, as well as a variety of manipulation and sensing heuristics to inform task planning. Furthermore, an original deformation modelling and prediction component (section III-B) is built to allow fast inference of the effect of candidate robot actions on the shape of the deformable object under manipulation.

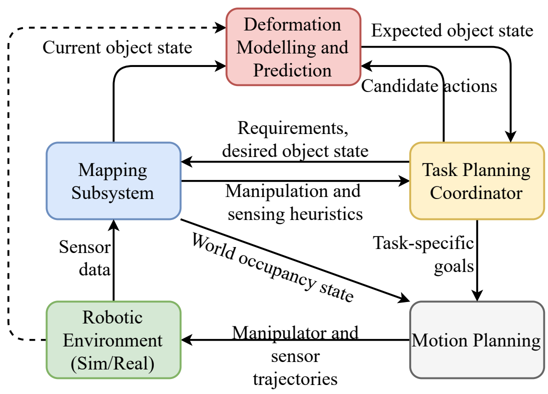

Fig. 2. Overview of the deformable object manipulation system architecture. The dotted line denotes the possibility to use the simulation ground-truth state directly for testing purposes. The target object shape and pose are defined by loading a map, e.g., from a mesh file or previous manipulation, and applying the desired transformations.

The heuristics from the mapping subsystem and the predictions of the deformation model are then used by a task planning coordinator, which follows the framework of [9], to generate task-specific goals that are passed to the motion planning and robotic components. In the cases considered for experimental evaluation in this paper, this task planning coordinator has two main roles. Firstly, it relies on the sensing heuristics to select viewpoints for the RGB-D camera to fully observe the initial shape of the object of interest and track it during manipulation. Secondly, it uses the manipulation heuristics to propose candidate manipulator actions that drive the shape and pose of the object of interest toward the target, validating them with the predictions of the deformation model.

Motion planning is implemented through the MoveIt package [10], which handles the generation of collision-free trajectories to reach the end-effector poses specified by the task planning coordinator. The robotic and sensing environment is built both in the real world and in simulation (section IV).

A. Mapping

The mapping subsystem is designed to be extensible and independent of the task and robotic platform used. The core of this system is built around efficient high-resolution voxel grids via the OpenVDB library [11], which encodes voxels in a shallow tree structure with a custom memory layout to achieve fast random access to the voxels as well as a small memory footprint. Important features of OpenVDB for this application include the representation of boolean, float or vector-valued voxel grids, efficient voxel iteration that skips "empty" voxels, as well as direct voxel access with Cartesian indexing in either grid or world units. This latter feature in particular allows converting well-established image processing algorithms to the 3D environment with relative ease.

To enhance reusability and compatibility with existing software, the mapping subsystem is designed as a self-contained ROS2 package [12] with parallel nodes that leverage the ROS2 message and service interfaces to enable map creation, update, querying and display by client applications. An overview of this package is shown in Fig. 3. The three nodes separate the concerns of creating/querying maps and next-best-view (NBV) heuristics (API node), updating them (Maintenance), and preparing different visualizations (Viewer) while the custom OpenVDB map repository handles the storage and synchronization of the maps.

Maps can be constructed with an arbitrary voxel size by taking as input a mesh file, a pointcloud stream from a ROS2 topic, the OctoMap planning scene published by a MoveIt node, or by applying an operation to an existing map. Map updates can be configured to happen upon receiving an update request, at a fixed interval, or automatically when the input data is updated. Pointcloud-based maps can capture occupancy data in a ternary or probabilistic manner, as well as colour or other measurement information. Occupancy data can also be output to a MoveIt topic to update the motion planning obstacles distribution. These different options create a mapping system that can be adapted to the task at hand as well as to the available computing resources.

Most important to the manipulation of a priori unknown deformable objects is the ability of the system to create maps by running arbitrary operations on existing maps, chaining and composing them to run complex operations with little effort and a minimal amount of intermediary maps. These

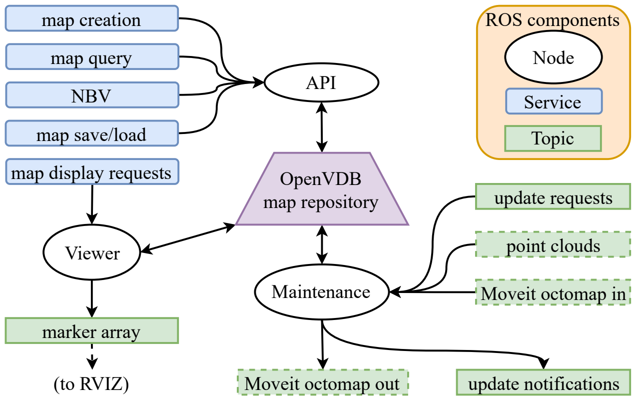

Fig. 3. Overview of the ROS2 mapping package.

operations are implemented as plug-in functions that operate within the mapping package, avoiding the transfer of maps to the client application. Various segmentation, tracking, conversion, clean-up, and shape comparison algorithms are implemented by way of these operations, enabling the construction of the different object representations and metrics needed to support the continuous reshaping of 3D deformable objects. Maps can then be queried to obtain, e.g., the values of arbitrary points, minimum- or maximum-valued regions, as well as sensors' next-best-view (NBV) heuristics for shape completion and object tracking.

The mapping subsystem is controlled by the task planning coordinator to create a set of 3D voxel maps, as shown in Fig. 4, which enables the definition of task-specific goals for observing and manipulating the shape of the object of interest. Point clouds received from the sensors are merged into a colour (RGB) map and an occupancy map. These maps are processed by a segmentation and tracking algorithm to extract the object of interest and distinguish it from the obstacles in the environment. The current object map is used for three purposes. First, next-best-view heuristics are computed to determine whether the current object map is likely to represent the entire shape of the object, and if not, where the camera should be positioned to observe the missing parts. Once the current object map is deemed to represent the entire object of interest, it is then compared with a predefined target object state to extract manipulation and reshaping heuristics that help in planning grasp points and robot motions. Finally, the current object map is used as a reference by the deformation model to determine the effect of candidate robot actions on the shape of the object.

B. Deformation Modelling

A deformation model is a critical system component for control and planning with deformable objects, as it provides a mechanism to request information about the expected state and shape of the object given an anticipated robot action. We build this component as an original dynamics model module that implements a function to transform robot control inputs to changes in the object state. This deformation modelling subsystem supports creating/querying different

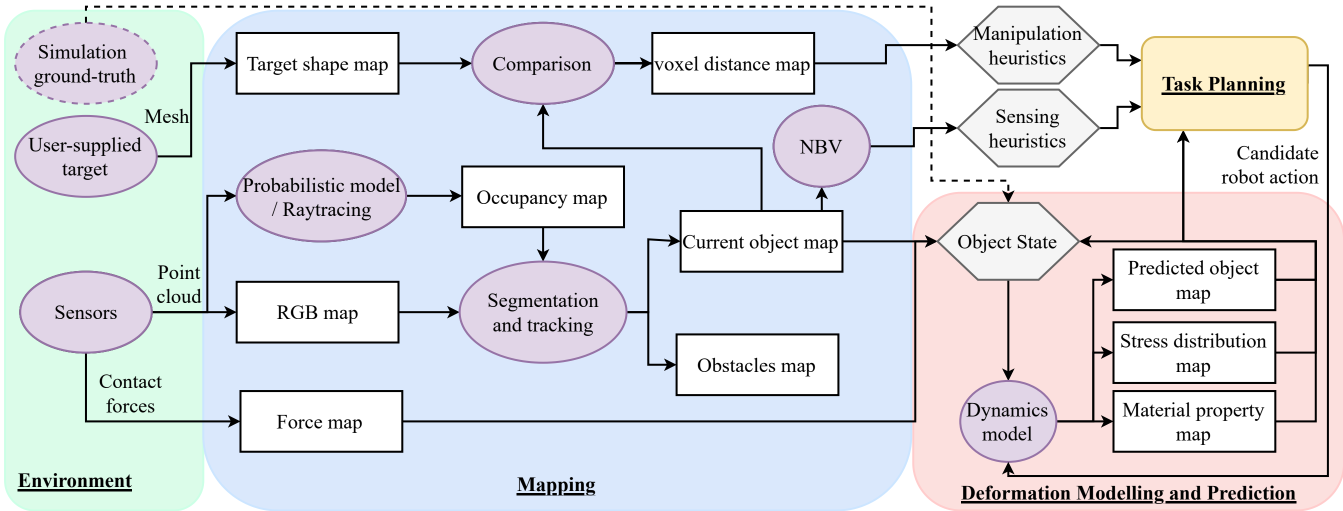

Fig. 4. Maps required for the 3D deformable object manipulation application. The dotted line denotes the possibility to use the simulation ground-truth state directly for testing purposes. Interactions with the robotic environment are shown in Fig. 2.

dynamics models depending on the available data, object state representation, and task specification.

The basis of this subsystem lies in the characterization of a predictive dynamics model [13]. In this way, it can be employed in real-time feedback control schemes for reshaping 3D deformable objects. In this configuration, the model allows continuous updates to the object state through rapid model evaluation and prioritizes short-term predictions, as sensor data is obtained in real-time. But more importantly, the dynamics model can be used for long-term predictions where information about future shape changes is obtained over time, which corresponds to a temporal description of the object state evolution. This process may involve not only a shape query from the object map, but also identifying the robot's contact locations on the object by means of a force map, which can be derived from tactile sensors or by simulating collisions between the gripper and the object.

Because the dynamics models receive the current state of the object and predict deformation quantities, they not only allow updating the object shape, but also support augmenting the state representation with physical descriptions, such as maps reflecting the material properties and internal stress distribution, offering additional information for the task at hand. These maps, heuristics and predictions can be integrated into manipulation planning algorithms to determine optimal action sequences. Moreover, they can be naturally combined with control algorithms to set the commands needed to achieve the desired configurations.

IV. Experimental Evaluation

The capabilities of the proposed system are tested in both simulation and real-world environments. First, the physical setup for validation is described, emphasizing the challenges of accurately measuring object deformation. This is followed by presenting the role of simulation and how it enables to accelerate the characterization of deformation models.

A. *Physical Setup and Challenges*

The robotic and sensing components were tested in the real world using a Kinova Jaco robot manipulator on a fixed base, customized to hold a RealSense D435i RGB-D camera near the end effector in an eye-in-hand configuration (Fig. 1). Working with this setup, a few significant limitations of the hardware were uncovered. In particular, the depth measurements offered by the camera have a relatively low accuracy, which impedes the shape analysis of small objects. While larger objects are less sensitive to this issue, they are difficult to position within the limited practical workspace of the robot where they could be manipulated and observed with the constraint of an initially unknown, partially observable environment.

Given these challenges and recent advances in the availability of GPU-accelerated dynamics modelling, which allow fast realistic simulation of large deformations on volumetric objects with integration of robotic contacts [8], we turn to simulation to test the integrated mapping, modelling, prediction and planning algorithms, circumventing the limitations of commercial sensors and hardware.

B. *Building 3D Deformable Models in Simulation*

Due to the lack of publicly available data to evaluate deformable object manipulation systems, a simulation API is built on top of robotic simulators to generate synthetic datasets of manipulation tasks with 3D deformable objects. This interface is designed to simplify the tedious and time-consuming processes of using simulators to create new scenes, including creating the robots, sensors and objects, as well as manually setting their parameters. The latter is particularly complex when dealing with 3D deformable objects, which require precise tuning of multiple parameters for simulating the deformation behaviour.

To address these issues, we build an interface to support the modular creation of simulation environments. These serve to evaluate the performance of the mapping and modelling components of the proposed software architecture, providing a benchmark for assessing their accuracy and effectiveness by simulating real-world scenarios. Compared to existing solutions, our interface introduces a higher level of abstraction and is specifically designed for the robotic manipulation of 3D deformable objects, allowing to configure the scene, access data and control the robot without having to dive into the details of the simulator to ensure its operation with deformation. It can run the same simulation setup with different physics backends, currently supporting PyBullet [14] and IsaacGym [15]. The physics backends support deformation via GPU-accelerated finite element method (FEM), which enables accurate simulation of soft-body dynamics, including contacts and sensor measurements. The interface enables access to detailed information on the manipulation process under various conditions, such as variations of initial shapes, contact locations, and applied forces.

These simulated environments allow the exclusion of certain modules to focus on the evaluation of specific software components. For example, sensing and state estimation modules may be ignored in order to assess the deformation modelling component. In this case, the simulated ground-truth is used directly to represent the complete observation of the object state, which can be used for the evaluation of dynamics models. Alternatively, simulated sensors can be accessed to evaluate the mapping component independently of the prediction module.

C. Experimental Results

Fig. 5 presents a sample of results from integrating the mapping, task planning, modelling and prediction modules with the simulated robotic environment, showing the ability of the system to coordinate the different steps of the process independently of the specific algorithms used within the modules. In Fig. 5a, a spherical object (either real or simulated) is observed by a mobile RGB-D camera to build a 3D colour voxel map, which is then segmented into an

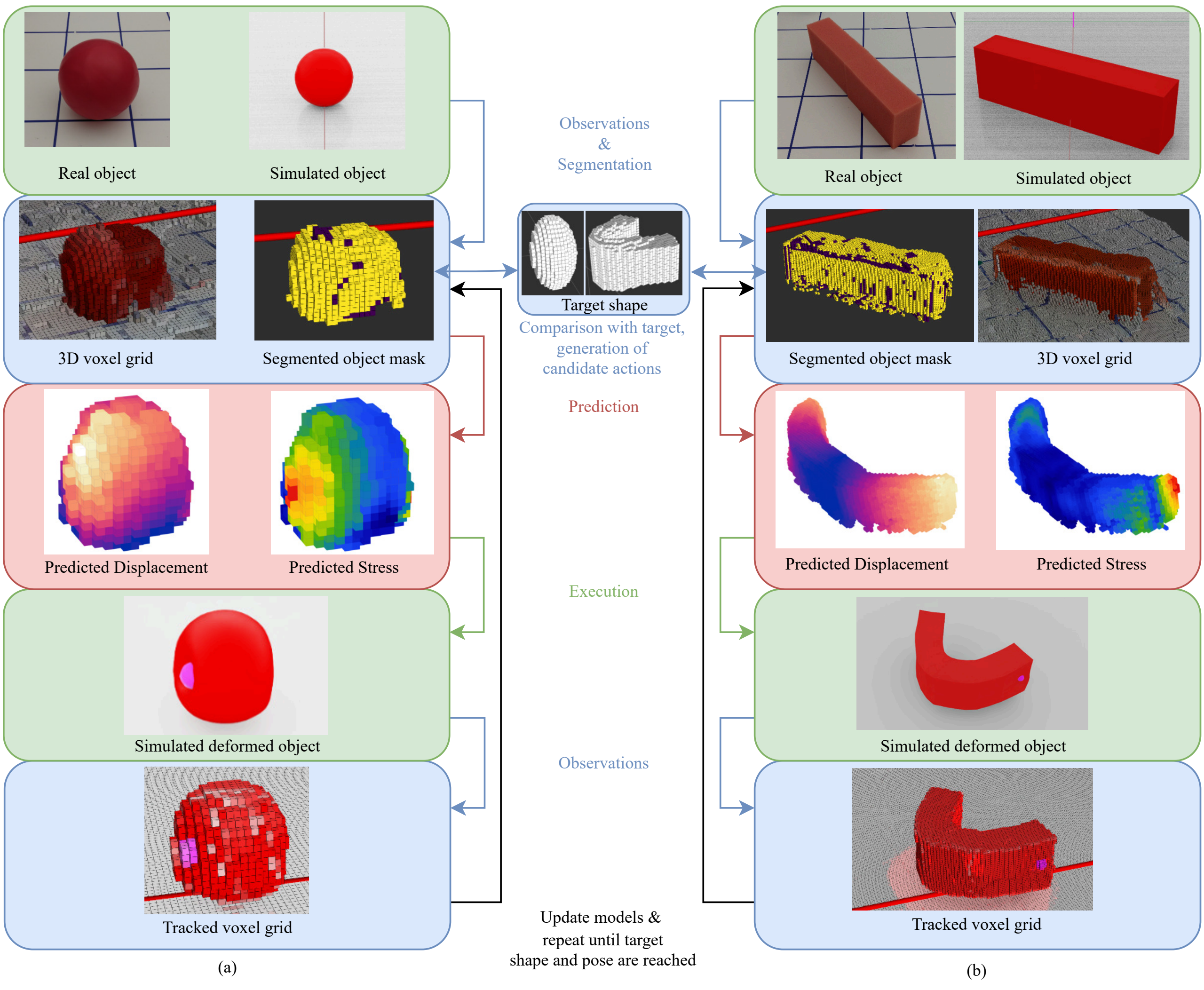

Fig. 5. Example manipulation of a spherical object (a) and rectangular object (b) with the proposed system.

object mask. This mask is compared with the target shape to generate potential actions and then used as an initial object shape for predicting the stress and displacement of the object under a proposed robot action, in this case squeezing two opposite sides of the sphere (Fig. 1). The selected action is executed (here in simulation, where the pink spheres simulate robot contacts) and the object shape is observed and tracked throughout the manipulation. Once the action is complete, the predictive model may be updated with the true behaviour of the object and the cycle is repeated until the desired object shape and pose are achieved. Fig. 5b shows the same process with a rectangular object that is bent into a "U" shape.

Comparing the initial 3D voxel grid to the physical objects (two upper rows in Fig. 5) shows the inaccuracy of the physical camera measurements, which may significantly misrepresent the shape of the object. With simulated measurements, the mapping subsystem is able to accurately represent and track the objects under manipulation. Computation time depends on experimental conditions such as the specific algorithms used within the modules, the size of the objects, and the resolution of the voxel grids. For the conditions shown in Fig. 5, almost all steps take fractions of a second, with bottlenecks being the planning and motion of the physical robot and the computation of the manipulation heuristics, which may each take up to a minute.

V. Conclusion

The intricacies of robotic interactions with deformable objects raise significant challenges in the design of general-purpose, reusable systems. This paper introduces a versatile, system-level conceptual framework to cope with the complex and high-dimensional problem of robotic manipulation on 3D deformable objects. A robotic system architecture is proposed which is composed of modular, task-agnostic components to achieve the reshaping and general manipulation of 3D deformable objects.

In particular, we present original mapping and deformation modelling modules which encapsulate the subtasks required for maintaining a map of the object and environment, extracting manipulation metrics, and predicting the effect of robot actions on the object's shape. In addition, a simulation interface is proposed to generate synthetic datasets for testing 3D deformable object mapping and modelling algorithms. With the addition of a task-specific planning coordinator and an off-the-shelf motion planning component, these modules form an end-to-end system. Experiments confirm the successful integration of the modules and the ability of the proposed system to efficiently coordinate the numerous tasks and components required for the manipulation and reshaping of initially unknown deformable objects in a partially observable environment.

Future work will be devoted to building an advanced task planner able to take full advantage of the proposed system for planning over long-horizon manipulation tasks with deformable objects in complex scenarios and with multiple sources of uncertainty.

Acknowledgements

The authors would like to acknowledge financial support to this research from the Natural Sciences and Engineering Research Council of Canada (NSERC) under the Discovery grants and ES-D scholarship programs, as well as from the Ontario Graduate Scholarship (OGS) program.

References

[1] F. Nadon, A. Valencia, and P. Payeur, "Multi-Modal Sensing and Robotic Manipulation of Non-Rigid Objects: A Survey," *Robotics*, vol. 7, no. 4, p. 74, Nov. 2018. [Online]. Available: http://www.mdpi.com/2218-6581/7/4/74

[2] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, Jun. 2018. [Online]. Available: https://doi.org/10.1177/0278364918779698

[3] R. Herguedas, G. López-Nicolás, R. Aragüés, and C. Sagüés, "Survey on multi-robot manipulation of deformable objects," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2019, pp. 977–984.

[4] M. Zürn, M. Wnuk, A. Lechler, and A. Verl, "Software Architecture For Deformable Linear Object Manipulation: A Shape Manipulation Case Study," in *2022 IEEE/ACM 4th International Workshop on Robotics Software Engineering (RoSE)*. New York, NY, USA: Association for Computing Machinery, May 2022, pp. 9–16.

[5] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013. [Online]. Available: http://link.springer.com/10.1007/s10514-012-9321-0

[6] S. Macenski, D. Tsai, and M. Feinberg, "Spatio-temporal voxel layer: A view on robot perception for the dynamic world," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, p. 1729881420910530, Mar. 2020. [Online]. Available: https://doi.org/10.1177/1729881420910530

[7] R. Antonova, P. Shi, H. Yin, Z. Weng, and D. K. Jensfelt, "Dynamic Environments with Deformable Objects," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1, 2021.

[8] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "ORBIT: A Unified Simulation Framework for Interactive Robot Learning Environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[9] F. Nadon, A. J. Valencia, N. Sambandam, S. Rowlands, J. Dickens, and P. Payeur, "Toward a General Framework for 3D Deformable Object Grasping and Manipulation," in *Workshop on Robotic Manipulation of Deformable Objects (ROMADO), IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020*, Oct. 2020, p. 4. [Online]. Available: http://commandia.unizar.es/irosworkshop2020/

[10] PickNick Robotics, "MoveIt Motion Planning Framework," 2024. [Online]. Available: https://moveit.ai/

[11] K. Museth, "VDB: High-resolution sparse volumes with dynamic topology," *ACM Transactions on Graphics*, vol. 32, no. 3, pp. 1–22, Jun. 2013.

[12] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074

[13] A. J. Valencia and P. Payeur, "Deformation Modeling for the Robotic Manipulation of 3D Elastic Objects using Physics-Informed Graph Neural Networks," in *20th Conference on Robots and Vision (CRV)*. Montreal, QC, Canada: IEEE, 2023, pp. 194–201.

[14] E. Coumans and Y. Bai, "PyBullet, a Python module for physics simulation for games, robotics and machine learning," 2016. [Online]. Available: http://pybullet.org

[15] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.