

# Cable Routing and Assembly using Tactile-driven Motion Primitives

Achu Wilson<sup>\*1</sup> Helen Jiang<sup>\*1</sup> Wenzhao Lian<sup>2</sup> Wenzhen Yuan<sup>1</sup>

**Abstract**—Manipulating cables is challenging for robots because of the infinite degrees of freedom of the cables and frequent occlusion by the gripper and the environment. These challenges are further complicated by the dexterous nature of the operations required for cable routing and assembly, such as weaving and inserting, hampering common solutions with vision-only sensing. In this paper, we propose to integrate tactile-guided low-level motion control with high-level vision-based task parsing for a challenging task: cable routing and assembly on a reconfigurable task board. Specifically, we build a library of tactile-guided motion primitives using a fingertip GelSight sensor, where each primitive reliably accomplishes an operation such as cable following and weaving. The overall task is inferred via visual perception given a goal configuration image, and then used to generate the primitive sequence. Experiments demonstrate the effectiveness of individual tactile-guided primitives and the integrated end-to-end solution, significantly outperforming the method without tactile sensing. Our reconfigurable task setup and proposed baselines provide a benchmark for future research in cable manipulation.

## I. INTRODUCTION

Robotic manipulation and planning has seen substantial progress in recent years. However, manipulating flexible objects such as cables and wires remains an open problem [1], [2], [3], [4]. In particular, tasks such as cable assembly require robot systems to be able to effectively track narrow cables and route them in between obstacles. Because the object of interest is thin, malleable, and subject to severe occlusions, this often requires advanced sensing, planning, and control.

While previous efforts in cord manipulation have considered knot tying [1], wire insertion [2], and (most relevant to us) cable routing [3], they are limited in task complexity and generalizability. Specifically, simplistic operations such as picking, moving, and placing prevent the robot from handling more complex tasks such as weaving parts through slots [5]. In addition, the above mentioned methods often cannot recover from failure due to the open-loop nature of the execution [3]. The main difficulty lies in how to continuously estimate the cable state, particularly when vision alone suffers from severe gripper/object occlusions, and how to generate motion commands accordingly.

Tactile-guided manipulation is a promising approach, but has only been demonstrated in a simple cable following task [6]. In this paper, we consider a full task of cable routing and assembly inspired by the NIST Assembly Task Board 3 [5]. We design a reconfigurable setup with fixtures that require

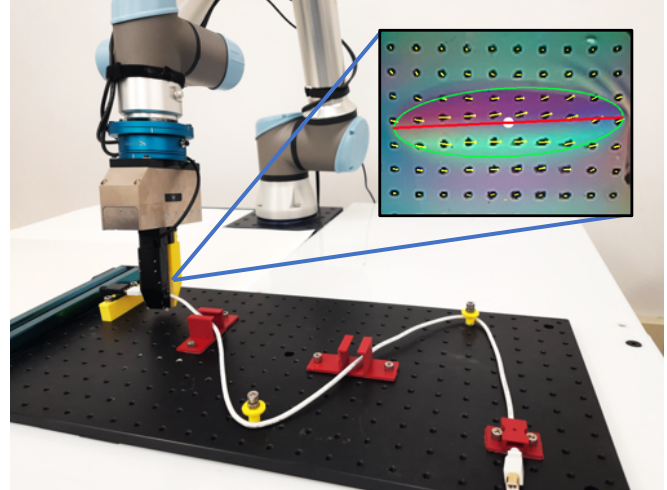


Fig. 1: Goal-conditioned cable manipulation using visual-tactile perception. Given an RGBD image of a goal configuration, our system parses the task, generates reference trajectories, and applies a sequence of tactile-guided motion primitives to accomplish a cable routing and assembly task.

4 operations covering common tethered object manipulation, as illustrated in Fig. 1. These tasks have longer horizons, and require reliable generalization across configurations.

To solve this problem, we propose to use visual perception for task understanding, i.e., a task description file is automatically extracted from a goal configuration image. The parsed task file is then mapped to a sequence of tactile-guided motion primitives. Notably, we leverage tactile sensing to design a library of primitives to continuously estimate the cable state and manipulate it in a closed-loop manner. Concretely, four primitives (cable following, pivoting, weaving, and insertion) are defined, each as a state machine, where the state transitions are governed by tactile perception. State-dependent controllers are then activated sequentially to realize the desired primitive behavior. Because of this modular design where a task is parsed via visual perception into primitives and each tactile-guided primitive is task context independent, our approach is able to generalize across different task board configurations with zero adaptation effort. Experiment results demonstrate the effectiveness and generalizability of the tactile-guided motion primitives, as well as our fully integrated pipeline.

Our contributions are summarized as follows. First, we propose a novel integrated solution for cable routing and assembly, where visual perception enables automatic high-level task parsing. Second, we design a library of tactile-

<sup>\*</sup>Authors with equal contribution.

<sup>1</sup>Carnegie Mellon University, PA, USA. {achuw, helenjia, wenzheny}@andrew.cmu.edu

<sup>2</sup>Intrinsic Innovation LLC, CA, USA. {wenzhaol}@google.com

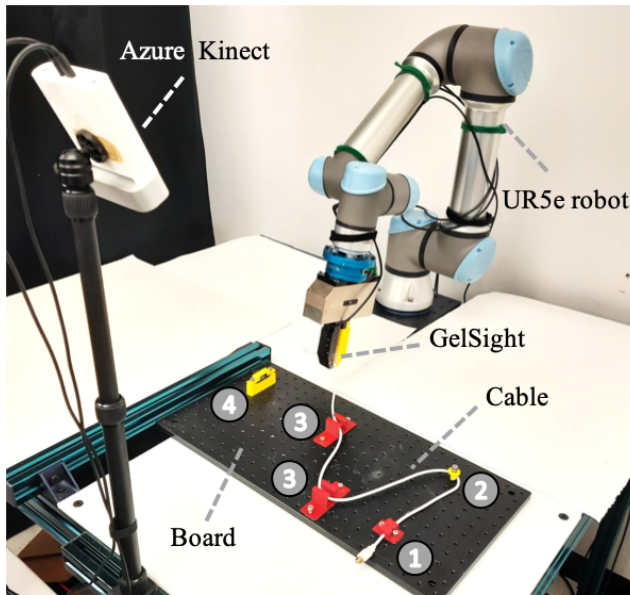


Fig. 2: **System setup and task board configuration.** The task board contains 4 types of fixtures, labeled 1-4 in the figure (1 = start, 2 = pivot, 3 = slot, 4 = USB connector). The cable begins at the start and ends being inserted into the USB connector. In between, the cable may be wrapped around the yellow pivots or woven through red slots. A UR5e robot with a GelSight sensor attached to the gripper is used to manipulate the cable. We also have a Azure Kinect camera overlooking the board to capture the board configuration.

guided motion primitives for low-level motion control to accomplish complex cable operations. Third, we provide baselines with and without tactile sensing on a reconfigurable cable routing and assembly task for future research.

## II. RELATED WORK

### A. Tactile sensing for robotics

The sense of touch, one of the main reasons that imparts dexterous and fine manipulation skills to human hands, has been inspiring robotics researchers since early days. A multitude of tactile sensing technologies [7] have been developed to aid robot manipulation. This includes measurement of features such as shape, texture, and forces. A recent and increasingly common mode of tactile sensing is optical tactile sensors such as GelSight [8] [9].

GelSight has been used for estimating contact shape and pose of objects [10] and manipulating them [11]. The shear and slip forces on a grasped object can also be estimated [12], which can be used to adjust the grasp [13]. [14] developed manipulation primitives that exploited tactile sensing for reactive manipulation of tabletop objects. [15] used tactile exploration to learn objects' physical properties for dynamic manipulation. [16] uses tactile sensing to estimate contact locations between a grasped object and the external environment for peg-in-hole insertion. End-to-end learning methods using tactile inputs have been developed for manipulation of small objects [17]. However, all of the above works focus on

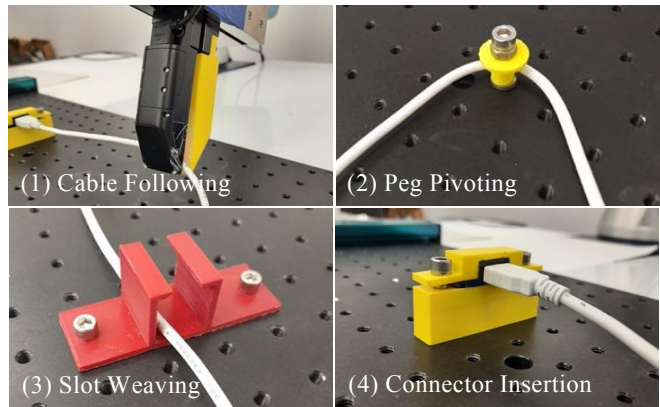


Fig. 3: **Task board operations.** Our task is split into 4 operations: 1) cable following (gripping and following along the cable), 2) peg pivoting (wrapping the cable around the vertical peg), 3) slot weaving (threading the cable through the horizontal slot), and 4) connector insertion (inserting the USB head at the end of the cable into the connector).

rigid object state estimation, while we investigate tasks that require continuous state estimation of deformable objects.

Another line of relevant work is tactile servoing. A neural network based edge detector taking tactile images as input was learned in [18], and was used for contour following. Reinforcement learning approaches for tactile contour following and closing a ziploc bag was developed in [19]. In the spectrum of classical control methods, a tactile servoing controller was designed to track an object edge as well as a cable lying flat on a surface [20]. In [21], a tactile-based closed-loop velocity controller was developed to regulate the sliding behavior of a grasped object. One of the closest work to ours is [6], where a GelSight gripper is utilized to enable a closed-loop controller for cable following in free space. In contrast, we tackle a more realistic scenario where the cables rest naturally on a task board, and a more complex task that involves routing and weaving in a constrained workspace.

### B. Deformable linear object manipulation and assembly

Manipulating deformable linear objects such as cables and ropes with robots is generally challenging due to the objects' infinite degrees of freedom. Previous works have attempted to design methods ranging from state estimation, representation learning, motion planning, to end-to-end learning.

In an early work [22], the cable deformation due to external forces is estimated using stereo vision, and manipulation techniques are developed to straighten the cable for through-hole insertion. [3] proposes a novel spatial representation between the cable and environment objects for motion planning. In recent years, end-to-end approaches were proposed to learn the deformation model from simulation and manipulate cables to a target shape [23]. These works commonly assume the system to be quasistatic and achieves manipulation using repetitive pick and place actions [24] [25]. Few approaches leverage environment contacts when manipulating cables, e.g., leveraging force-torque sensing

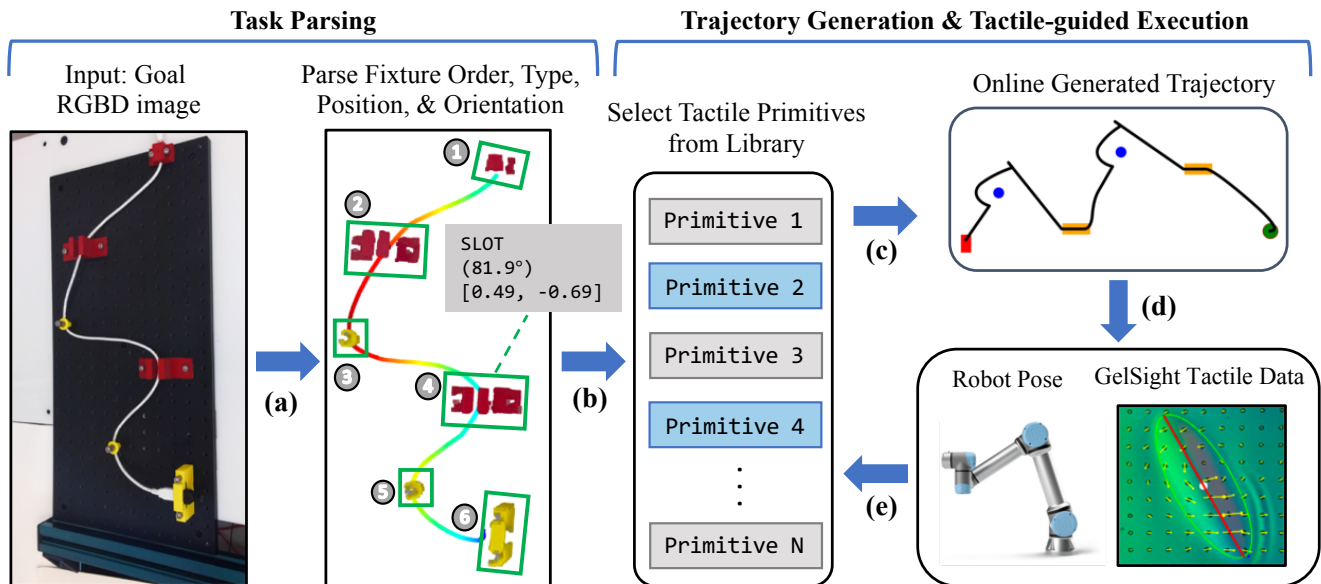


Fig. 4: **Pipeline overview.** We propose an end-to-end framework for cable routing and assembly. (a) Our vision module takes in RGBD images of the goal taskboard with and without the cable to output the task specification. We use color filtering to determine the fixtures’ positions and types, Principal Component Analysis (PCA) to determine the orientations and shapes, and Coherent Point Drift (CPD) to detect the cable thus determining the fixture order. (b) The parsed task description (fixture order, type, position and orientation) is mapped to a sequence of parameterized tactile primitives, meanwhile generating an initial reference robot trajectory. (c, d, e) Sequentially, the robot executes each primitive as an individual state machine, where the state transitions are governed by the sensed tactile data. In each state, a parametric trajectory generator is activated to generate the trajectory online, for example, with lines and splines.

[26] or visual perception [27]. [28] proposes a task-space planner, which builds a roadmap from predefined tasks and employs a replanning strategy based on a genetic algorithm which executes cable routing using a dual-arm robot. [29] developed a system for insertion of wire-terminal insertion via visuo-tactile methods. [30] developed Bayesian state estimation methods to predict symbolic states with predicate classifiers for connector insertion. In this work, we aim to build an entire cable routing and connector insertion system using visual sensing for initial plan generation and then tactile perception to monitor the cable-environment contact state, and adjust the control policy accordingly.

### III. PROBLEM STATEMENT

#### A. Task

We aim to solve the cable manipulation problem inspired by the NIST Assembly Task Board 3[5]. We consider a taskboard that consists of fixtures such as pegs and channels to route and manipulate the cable to a pre-specified configuration. The taskboard consists of 4 different types of fixtures in various configurations: 1) Start fixture, 2) Vertical pegs / pivots, 3) Horizontal slots, and 4) USB connector. An example of the taskboard with fixtures with its goal configuration is shown in Fig. 2.

#### B. Operations

To move the cable around the fixtures on the board, we divide the entire task into the following subtasks, as illustrated in Fig. 3.

1) *Cable following:* The robot holds the cable and follows along without dropping it.

2) *Pivoting around vertical pegs:* The robot pivots the cable’s heading direction by rotating it around the pegs.

3) *Weaving through horizontal slots:* The robot weaves or threads the cable through the horizontal channel slots that “lock” the cable inside.

4) *Connector insertion:* The robot inserts the USB head at the cable’s end into the connector fixture.

### IV. METHOD

The proposed method starts with estimating the configuration of the taskboard and the goal cable configuration from an assembled taskboard. Once the taskboard configuration is estimated, a task description is generated and mapped to a sequence of parameterized tactile primitives. Each primitive is designed as an individual state machine, where the state transitions are governed by the sensed tactile data. The robot executes primitives sequentially until task completion.

#### A. Perception systems & task parsing

The perception system uses RGBD data from a Microsoft Azure Kinect camera overlooking the task board and tactile data generated by a GelSight R1.5 optical tactile sensor.

1) *Visual perception:* The visual perception module requires an assembled task board from human demonstration, and infers the goal configuration of the cable. Given a point cloud from the Kinect camera, we aim to recover the type, position, orientation and ordering of each fixture on the task

board. We also track the demonstration cable in the image to parse the order of the fixtures along it. The task parsing section in Fig. 4 shows an example input-output pair of the visual perception module.

**Fixture pose estimation:** We first use a point cloud of the peg board with the fixtures but without the cable. We locate all fixtures via simple color filtering and clustering. We apply Principal Component Analysis (PCA) to estimate the type (using the shape, or oblongness) and orientation of each detected fixture. The “start” and “connector” fixtures needs their orientation ambiguity to be resolved, so we use the cable heading direction to infer their orientations. Specifically, we take the point from the cable leading up to the given fixture and the current point of the cable the fixture is on to calculate the directional vector.

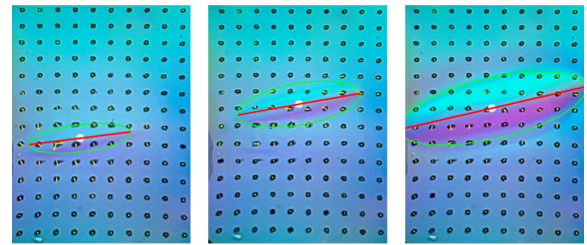
**Cable state estimation:** We track the cable in the demonstration board to get the correct order of the routing task. To segment the points corresponding to the white cable, we take all of the white-colored points from the point cloud from the second RGBD image defining the goal cable configuration. The resulting cable point cloud has occlusions created by the fixtures, so we use Reeb Graph [31] to construct a set of cable nodes. This gives an initialization for Coherent Point Drift (CPD) [32] to complete the cable. With the completed sequence of cable nodes, the order of the fixtures are readily determined by tracing the cable. This cable node sequence also disambiguates the orientation of the “start” and “connector” fixtures, as described in the paragraph above.

Using this visual perception pipeline, the type, location, orientation and order of the fixtures are estimated, which defines the routing and assembly task. This task description then serves as the input for the motion primitives introduced in Section IV-B.

2) *Tactile system:* We use the tactile reading to estimate the cable state in different stages and adjust the manipulation strategies accordingly. The fingertip GelSight sensor provides tactile images of the contact area and marker displacement information corresponding to the 3-axis forces and in-plane torque on the contact surface. As introduced in [8], the change of the color in the GelSight images corresponds to the contact geometry and the motion of the markers in the images indicates the contact force and torque: a “spreading out” pattern of the markers’ motion indicates normal force, a uniform motion pattern of the markers indicates shear force towards the motion direction, and a spiral pattern indicates an in-plane torque. The magnitude of the marker’s motion is approximately linear to the magnitude of the force.

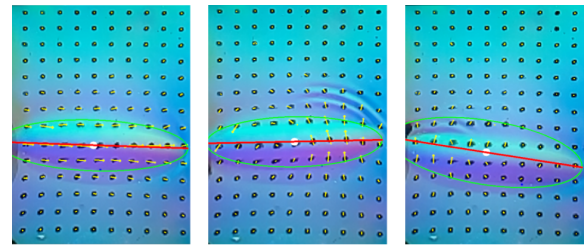
In this work, we estimate the contact area of the cable based on color and background image subtraction. The contact area of the cable is elliptical in shape, as shown in Fig. 5. The area also corresponds to the normal force, which is used to detect the firmness of the cable gripping. We fit an ellipse to the contact area and use its center and major axis to estimate the pose of the cable in hand. This helps the robot to re-center and re-orient the cable in the gripper.

Force and torque changes from the markers are tracked to identify cable states, such as a slowly increasing shear force



(a) Grip Force = 5N (b) Grip Force = 10N (c) Grip Force = 25N

Fig. 5: When holding the cable with different forces, the contact area measured from GelSight is an elliptical shape. The area of the ellipse is roughly linear to the gripping force, and its center and orientation help estimate the cable pose.



(a) Shear Force (b) CCW Torque (c) CW Torque

Fig. 6: **Motion patterns of the GelSight markers.** We use the motion patterns of the GelSight markers to determine contact forces and torques which are used to estimate the cable states in different stages. (a) Shear forces along the cable indicate tension when dragging the cable. (b) A counterclockwise (CCW) torque is generated when the USB cable head hits the connector fixture from the top. (c) A clockwise (CW) torque is produced when the cable rear makes contact with a slot fixture from the top.

along the cable indicating tight cable hold, and fast changing contact force/torque indicating a collision between the cable and the fixture. Fig. 6 shows some example tactile images used for state estimation at different stages.

The GelSight marker magnitudes also serve as input for the hybrid force-position controller used in connector insertion. GelSight data is sampled at 60 Hz with a latency of around 75 ms, while the force-position controller is a cascaded PID controller with an update frequency of 250 Hz. To match the controller frequency, the low frequency GelSight data is linearly interpolated in time. To mitigate the destabilizing effects of latency, high-level commands are issued to the controller at a considerably lower frequency than the control loop’s operating frequency.

### B. Motion primitives and trajectory generation

The vision system infers an ordered list of fixture type, position, and orientation for task specification. This information is utilized to generate a reference robot path, as illustrated in the Online Generated Trajectory subplot in Fig. 4, which is divided into sections for each fixture. Motion primitives designed for each task are executed for the corresponding path sections. The robot begins sequentially sets the end

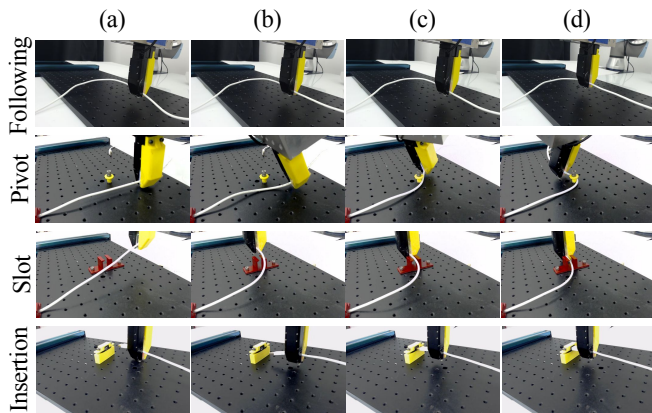


Fig. 7: **Motion primitives.** To complete our cable routing and assembly task, we construct a library of motion primitives. The cable following primitive moves the robot along the cable, while pulling it in a desired direction. The pivot primitive wraps the cable around a peg while maintaining cable tension. The slot primitive inserts the cable into the slot and “locks it” in place. The insertion primitive aligns and inserts the USB connector into the socket.

position of the previous primitive as the starting position for the next until the task is completed.

The primitives are parameterized and modeled from observing a human performing the task and deriving heuristics. The primitives are designed as state machines, whose state transitions are triggered by the tactile signals generated by the external forces on the cable, as shown in Fig. 7. Following are the four primitives:

1) *Cable following:* The robot performs this primitive to guide the cable from one fixture to another. As illustrated in the first row of Fig. 7, the primitive executes the following state sequence in a loop: i) the gripper slowly closes until the gripping force detected by GelSight exceeds a threshold; ii) the robot pulls the cable until the sensed shear force exceeds a threshold, indicating the cable tension; iii) the gripper slowly opens until the gripping force falls below a threshold; iv) based on the detected cable pose in the gripper, the robot slides along the cable while keeping the cable centered.

2) *Pivoting around pegs:* The pivoting primitive changes the direction of the cable to make it pivot around a vertical peg, as shown in the second row in Fig. 7. This primitive starts with following along the cable until a waypoint, determined by the peg position. The robot tilts backwards and moves down until the tactile signals indicate the cable is in contact with the taskboard, as illustrated in Fig. 7 top row, plot (b). These tactile signals look similar to the clockwise torque signals in Fig. 6 (c), ensuring that the cable gets tucked under the pivoting peg. The robot then moves in a circular trajectory centered at the initial position, and continues until the cable makes contact with the peg as in plot (c). After establishing contact between the cable and peg, a new circular trajectory is generated using the peg’s position as the center.

3) *Weaving through slots:* The weaving primitive is used to guide the cable through horizontal slots, as shown in the third row of Fig. 7. It is parameterized by the slot position and orientation, which are used to generate a reference trajectory to place the cable into the slot from the top. If the cable is not aligned with the slot during the downward motion, as indicated by a sudden increase of the in-plane torque from the GelSight image, the robot executes a horizontal wiggling action while moving down. It is continued until the in-plane torque disappears, signaling that the collision is resolved and the cable is aligned with the slot.

4) *Connector insertion:* This primitive conducts tethered USB connector insertion given the socket position and orientation, as shown in the bottom row of Fig. 7. The primitive starts from a grasp on the cable near the connector, which enables a certain amount of freedom for the connector to bend when hitting the rigid side of the socket fixture. When the connector contacts the socket fixture, the GelSight sensor detects a normal force along the cable. We then command the robot using a hybrid force-position controller to rotate around the normal axis while keeping the normal force. The motion enables the connector to move for a small distance around the initial contact location. When the connector falls into the socket, there will be a sudden drop in the normal force on the cable and a strong torque that stops the cable from continuing rotation. Both the change of force and torque can be detected by GelSight, and we will make the robot stop this “exploration” procedure and push forward to insert the connector into the socket. In some cases, if the initial contact point is too far away from the socket, the rotary exploration motion will not get the connector inside the socket. We will then make the robot retreat and start the exploration from another randomly-sampled location near the socket.

## V. EXPERIMENTS

In this section, we evaluate the performance of our cable routing and assembly pipeline. Specifically, we investigate the benefits of tactile sensing in cable manipulation tasks and the reliability of our designed tactile primitives.

### A. Robot setup

Our full setup is shown in Fig. 2. We use a UR5e 6-DoF robot arm switching between position control and hybrid force-position control modes. To grasp the cable, we use a Weiss Robotics WSG-50 gripper, which operates in an indirect force control. Mounted on the gripper is a GelSight R1.5 sensor along with a custom designed and 3D printed finger which has design features to prevent accidental cable drops. These design features include an inner surface with an elastomer pad with adequate friction, a shape which bulges slightly towards the tip, and a beak on one side which helps picking up thin cables from the flat surface and holds it from falling down. We also use a Microsoft Azure Kinect camera to capture the entire taskboard and acquire RGBD data for task parsing.

Primitive	Cable Following	Peg Pivoting	Slot Weaving	Connector Insertion
w/o Tactile (Baseline)	43/50	28/50	32/50	0/50
w/ Tactile (Ours)	<b>50/50</b>	<b>50/50</b>	<b>50/50</b>	<b>48/50</b>

TABLE I: **Success rate of individual primitives.** For each primitive, we calculate the success rate with and without tactile sensing.

	Success Rate	Failure Modes
GT + Tactile (Ceiling)	53/60	A(3), B(2), C(2), D(0)
GT + No Tactile (Baseline)	0/60	A(0), B(2), C(10), D(48)
Vision + Tactile (Ours)	<b>51/60</b>	A(4), B(3), C(2), D(0)

TABLE II: **Success rate of entire pipeline.** We summarize the success rate of the whole task across 6 different board configurations with 10 trials each. Failure Modes: A (Connector insertion failed), B (Cable got stuck), C (Cable escaped from previous fixture), D (Accumulated error in motion)

### B. Motion primitives with and without tactile sensing

We demonstrate the robustness of the tactile-guided motion primitives by comparing the success rate against a baseline without tactile sensing. The baseline applies the same trajectory which is generated analytically and does not use tactile sensing for adapting it.

We run 50 trials for each primitive (see Fig. 7) with varying fixture positions, and present the results in Table I. A trial is considered success if the robot is able to properly perform the corresponding primitive on that fixture. For all four primitives, our tactile-guided approach significantly outperforms the baseline without tactile sensing. The connector insertion primitive shows this the most clearly: the baseline has a success rate of 0% because the robot cannot align the orientation of the USB head with the connector, whereas with the tactile sensing, the controller can reorient the connector pose to align with the socket. The other baseline primitives fail due to the inability to estimate contacts and maintain tension in the cable.

### C. End-to-end cable routing and assembly

We analyze the reliability of our full pipeline over various board configurations. First to evaluate the performance of our vision pipeline described in Section IV-A.1, we compare our approach (Vision + Tactile) against an oracle baseline that uses the ground truth (i.e. human-specified) board configuration (GT + Tactile). We consider this to be an upper bound on performance. To evaluate the effectiveness of using tactile sensing, we experimented with a baseline without tactile sensing but using the ground truth board configuration (GT + No Tactile).

We compare the success rate and failure modes by evaluating each approach across 6 different board configurations (Fig. 8), with 10 trials each, summarized in Table II. We label the trial as a success if the entire task was completed. The oracle which uses ground truth board configurations

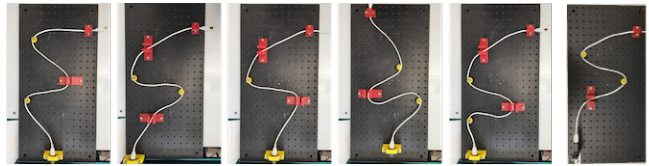


Fig. 8: **Taskboard configurations.** We evaluate cable routing and assembly on each of these 6 boards 10 times.

(53/60) performs only slightly better than our Vision + Tactile pipeline (51/60), suggesting that our vision pipeline can effectively parse the board task configuration. We find that the baseline without tactile sensing fails catastrophically due to the accumulated motion error, typically after the second or third fixture. While individual primitives could be completed without tactile sensing, although with limited success, as shown in Table I, we find that chaining the primitives reliably is difficult. This happens due to the lack of tactile feedback which corrects the trajectory while following the parsed path. The robot may also experience failures beyond its direct control, such as cable entanglement in fixtures or cable detachment from previously completed fixtures. During complex operations such as pivoting, weaving, and insertion, tactile signals are crucial for inferring the cable-environment interaction state to online adapt the trajectory. As summarized in the failure modes, our method occasionally fails because of the cable getting stuck or slack, and connector insertion remains the most challenging operation. We postulate such “global” state changes cannot be sufficiently captured by “local” tactile signals. Thus we plan to fuse the global visual sensing with local tactile sensing in the future to further improve the reliability.

## VI. CONCLUSION AND FUTURE WORK

This paper studies a long-horizon task that involves cable routing and tethered object assembly. Particularly, we propose a novel integrated pipeline where a task is parsed via visual perception and trajectories are planned and executed in a closed-loop manner with tactile-guided motion primitives. We compare the designed method and other baselines on a reconfigurable task board, which may serve as a benchmark for future research. Experiment results indicate the necessity of tactile sensing in such tasks involving dexterous operations with cables in a realistic environment.

One key limitation of our work is that the primitives are hand-designed rather than learned from data, thus limiting their generalizability. It is interesting to create more diverse and powerful primitives leveraging imitation learning and reinforcement learning. Another limitation is the visual and tactile perception being used at different stages. In future work, we plan to fuse the sensor modalities, joining global and local information, and learning task-relevant state representations.

## REFERENCES

- [1] J. Grannen, P. Sundaresan, B. Thananjeyan, J. Ichnowski, A. Balakrishna, V. Viswanath, M. Laskey, J. Gonzalez, and K. Goldberg,

- “Untangling dense knots by learning task-relevant keypoints,” in *Conference on Robot Learning*. PMLR, 2021, pp. 782–800.
- [2] S. Pirozzi and C. Natale, “Tactile-based manipulation of wires for switchgear assembly,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2650–2661, 2018.
- [3] S. Jin, W. Lian, C. Wang, M. Tomizuka, and S. Schaal, “Robotic cable routing with spatial representation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5687–5694, 2022.
- [4] K. Galassi and G. Palli, “Robotic wires manipulation for switchgear cabling and wiring harness manufacturing,” in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 531–536.
- [5] “<https://www.nist.gov/el/intelligent-systems-division-73500/iros-2020-robotic-grasping-and-manipulation-competition>.” [Online]. Available: <https://www.nist.gov/el/intelligent-systems-division-73500/iros-2020-robotic-grasping-and-manipulation-competition>
- [6] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, “Cable manipulation with a tactile-reactive gripper,” *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.
- [7] C. Chi, X. Sun, N. Xue, T. Li, and C. Liu, “Recent progress in technologies for tactile sensors,” *Sensors*, vol. 18, no. 4, 2018. [Online]. Available: <https://www.mdpi.com/1424-8220/18/4/948>
- [8] W. Yuan, S. Dong, and E. H. Adelson, “Gelsight: High-resolution robot tactile sensors for estimating geometry and force,” *Sensors*, vol. 17, no. 12, p. 2762, 2017.
- [9] I. H. Taylor, S. Dong, and A. Rodriguez, “Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10781–10787.
- [10] M. Bauza, O. Canal, and A. Rodriguez, “Tactile mapping and localization from high-resolution tactile imprints,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3811–3817.
- [11] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson, “Localization and manipulation of small parts using gelsight tactile sensing,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3988–3993.
- [12] S. Dong, W. Yuan, and E. H. Adelson, “Improved gelsight tactile sensor for measuring geometry and slip,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 137–144.
- [13] F. R. Hogan, M. Bauza, O. Canal, E. Donlon, and A. Rodriguez, “Tactile regrasp: Grasp adjustments via simulated tactile transformations,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2963–2970.
- [14] F. R. Hogan, J. Ballester, S. Dong, and A. Rodriguez, “Tactile dexterity: Manipulation primitives with tactile feedback,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8863–8869.
- [15] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, “Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5633–5640.
- [16] S. Kim and A. Rodriguez, “Active extrinsic contact sensing: Application to general peg-in-hole insertion,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10241–10247.
- [17] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine, “Manipulation by feel: Touch-based control with deep predictive models,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 818–824.
- [18] N. F. Lepora, A. Church, C. de Kerckhove, R. Hadsell, and J. Lloyd, “From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2101–2107, 2019.
- [19] R. B. Hellman, C. Tekin, M. van der Schaar, and V. J. Santos, “Functional contour-following via haptic perception and reinforcement learning,” *IEEE Transactions on Haptics*, vol. 11, no. 1, pp. 61–72, 2018.
- [20] Q. Li, C. Schürmann, R. Haschke, and H. J. Ritter, “A control framework for tactile servoing,” in *Robotics: Science and systems*. Citeseer, 2013.
- [21] Y. Chen, C. Prepscius, D. Lee, and D. D. Lee, “Tactile velocity estimation for controlled in-grasp sliding,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1614–1621, 2021.
- [22] H. Nakagaki, K. Kitagaki, T. Ogasawara, and H. Tsukune, “Study of deformation and insertion tasks of a flexible wire,” in *Proceedings of International Conference on Robotics and Automation*, vol. 3, 1997, pp. 2397–2402 vol.3.
- [23] C. Wang, Y. Zhang, X. Zhang, Z. Wu, X. Zhu, S. Jin, T. Tang, and M. Tomizuka, “Offline-online learning of deformation model for cable manipulation with graph neural networks,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5544–5551, apr 2022.
- [24] M. Yan, Y. Zhu, N. Jin, and J. Bohg, “Self-supervised learning of state estimation for manipulating deformable linear objects,” *IEEE robotics and automation letters*, vol. 5, no. 2, pp. 2372–2379, 2020.
- [25] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, “Combining self-supervised learning and imitation for vision-based rope manipulation,” in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2146–2153.
- [26] F. Süßerkrüb, R. Laezza, and Y. Karayiannidis, “Feel the tension: Manipulation of deformable linear objects in environments with fixtures using force information,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11216–11222.
- [27] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, “Robotic manipulation planning for shaping deformable linear objects with environmental contacts,” *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 16–23, 2020.
- [28] G. A. Waltersson, R. Laezza, and Y. Karayiannidis, “Planning and control for cable-routing with dual-arm robot,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1046–1052.
- [29] D. De Gregorio, R. Zanella, G. Palli, S. Pirozzi, and C. Melchiorri, “Integration of robotic vision and tactile sensing for wire-terminal insertion tasks,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 585–598, 2019.
- [30] T. Migimatsu, W. Lian, J. Bohg, and S. Schaal, “Symbolic state estimation with predicates for contact-rich manipulation tasks,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1702–1709.
- [31] J. Schulman, A. Lee, J. Ho, and P. Abbeel, “Tracking deformable objects with point clouds,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1130–1137.
- [32] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.