ICRA DLO Dataset Papers

Todo:

- [x] Find ICRA dataset papers
- [ ] Summarize papers
- [ ] Work on the Related Works section
- [ ] Mind-Map paper structure
- [ ] Mind-Map research threads and methods

Project Scope

ICRA DL Dataset Paper Structure

- Abstract
- Introduction
- Related Work
    - Perception in XYZ Application
    - XYZ Sense Modality Fusion
    - Grasping with Application-Specific Literature
- Proposed Method
    - Problem Statement: Usually, need for a trained model and dataset for a specific application.
    - Model Description: An application-specific trained model, usually based on a well-known model. Sometimes multiple models are combined to form a system or a larger model.
- Experiments
    - The Dataset: Based on another dataset or novel. Technical details on robotic and training hardware, sensors and data capture configurations.
    - Performance and Results: Define metrics, and identify factors that directly affect performance for each task (e.g. input resolution, sequence length on detection, tracking and etc).
    - Confusion matrix
    - Other Unique Experiments and Results
- Conclusion and Future Work

Papers

Other Papers

- 1. Simulation Tools for Model-Based Robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX

2022

- [ ] 1. (non-ICRA, HAL) Tactile Classification of Object Materials for Virtual Reality based Robot Teleoperation
- [ ] 2. Tracking Multiple Zebrafish Larvae Using YOLOv5 and DeepSORT
- [ ] 3. Shape Control of Deformable Linear Objects with Offline and Online Learning of Local Linear Deformation Models

2021

- [x] 1. Deformable Linear Object Prediction Using Locally Linear Latent Dynamics
- [ ] 2. Robots of the Lost Arc: Self-Supervised Learning to Dynamically Manipulate Fixed-Endpoint Cables
- [ ] 3. Learning to Rearrange Deformable Cables, Fabrics, and Bags with Goal-Conditioned Transporter Networks
- [ ] 4. Sim-to-Real for Robotic Tactile Sensing via Physics-Based Simulation and Learned Latent Projections
- [ ] 5. ReForm: A Robot Learning Sandbox for Deformable Linear Object Manipulation
- [ ] 6. Learning to Propagate Interaction Effects for Modeling Deformable Linear Objects Dynamics
- [ ] 7. NeuralSim: Augmenting Differentiable Simulators with Neural Networks
- [ ] 8. Real-to-Sim Registration of Deformable Soft-Tissue with Position-Based Dynamics for Surgical Robot Autonomy
- [ ] 9. DOT: Dynamic Object Tracking for Visual SLAM
- [ ] 10. SD-DefSLAM: Semi-Direct Monocular SLAM for Deformable and Intracorporeal Scenes
- [ ] 11. VelocityNet: Motion-Driven Feature Aggregation for 3D Object Detection in Point Cloud Sequences
- [ ] 12. Auto-Tuned Sim-to-Real Transfer
- [ ] 13. Tracking Partially-Occluded Deformable Objects while Enforcing Geometric Constraints
- [ ] 14. TSDF++: A Multi-Object Formulation for Dynamic Object Tracking and Reconstruction
- [ ] 15. Uncertainty-aware Self-supervised Target-mass Grasping of Granular Foods
- [ ] 16. Uncertainty-Aware Fast Curb Detection Using Convolutional Networks in Point Clouds
- [ ] 17. Hierarchical Planning for Long-Horizon Manipulation with Geometric and Symbolic Scene Graphs

2020

- [x] 1. Grasp State Assessment of Deformable Objects Using Visual-Tactile Fusion Perception

2018

- [ ] 1. Robotic Manipulation and Sensing of Deformable Objects
- [ ] 2. EMD Net: An Encode–Manipulate–Decode Network for Cloth Manipulation

2017

- [ ] 1. Combining Self-Supervised Learning and Imitation for Vision-Based Rope Manipulation

Summaries

ICRA 2022

Tactile Classification of Object Materials for Virtual Reality based Robot Teleoperation

Skip, since it is not an ICRA paper. It is a good technical paper.

Tracking Multiple Zebrafish Larvae Using YOLOv5 and DeepSORT

ICRA 2021

001 - Deformable Linear Object Prediction Using Locally Linear Latent Dynamics

- Brief: A framework for DLO prediction. Use DL to map DLO's non-linear dynamics to a linear space to make predictions. Their model is a locally-linear, action-conditioned dynamics model that is used as a predictor. They also use a \emph{sampling-based optimization} algorithm to select the \emph{optimal control action}. They empirically demonstrate model accuracy up to 10 states into the future. They can also find optimal action given an initial state and a goal state.
- Model: repo
    - Prediction Framework:
        - A state autoencoder: based on CNNs
        - An action autoencoder
        - A dynamic model
- Intro:
    - Prediction model: Use autoencoder to encode both the state and the action into latent state and latent action. Then, they run the dynamics model to get (the state matrix and control matrix for the linear dynamics model in the latent space). Now, they have the current latent state, latent action, state matrix and control matrix. They obtain a predicted next latent state by running the locally linear latent dynamics model with the mentioned state matrices and vectors. In the last step, they decode the predicted latent next state to a predicted next state. They can predict up to the next 10 future states.
    - Optimal Control Action Selector: They use a sampling-based optimization method to converge on an optimal action sequence based on given an initial condition state, a goal state, and the trained network.
- Contributions:
    - Prediction Model: An autoencoder-decoder-based model for DLO dynamic prediction.
    - Control Model: An sampling-based optimization control algorithm. Experiment and show comparative results.
    - Ablation studies on locally and globally latent dynamics.
- Related work: see mind-map.
- Method:
    - Preliminary
        - Rope Dynamics
        - Latent Dynamics
        - Encoder Model
        - Decoder Model
        - Dynamics Model
    - Prediction Framework
        - Loss function
        - Training prediction framework

        - Training autoencoders
        - Training Dynamics model
    - Sampling-Based model predictive control: used cross-entropy method (CEM) [31]

- Experiments and Results:

    - Experimental Setup:
        - Dataset: They used Rope Manipulation Dataset [26] (images are reduced from 240x240 to 50x50)
        - State: they augmented the data with translation, vertical flips and horizontal flips.
        - Action: XY pose in image space, l moving length, \theta moving angle in world space.
    - Multi-step Prediction Results and Comparison:
        - One step prediction
        - Multi-step prediction
        - Comparison: E2C used optimal control formulation in the latent space but the authors use the dynamics model to get the state matrix for state prediction in latent space.
    - Action Planning and Control Results
        - Qualitative results
        - Quantitative results: they use MSE between sampled actions and GT actions. They ran 100 single-step actions to get the mean and STD.
    - Ablation Study:
        - Locally latent dynamics vs globally latent dynamics: local dynamics, the state matrix and the control matrix are different for each step but it is more an accurate predictor.
        - Order for Training Models:
            1. Train autoencoders then dynamics model (better and more accurate)
            2. Train autoencoders and dynamic model at the same time (blurry rope images)
        - Latent state and latent action size: x is 50x50, experiments show the latent state should be at least 50 in size and latent action size should be at least 10. They chose 80 and 80.
        - Whether using action autoencoder: Two contracting statements??

- Sensors: RGB Images

- Hardware:

- Sensor-Fusion:

- Dataset details:

- Additional details:

002 - Robots of the Lost Arc: Self-Supervised Learning to Dynamically Manipulate Fixed-Endpoint Cables

- Abstract: In this paper, the authors focus on high-speed robot arm motions with ropes and cables. They teach a UR5 arm to vault over obstacles, knock objects from pedestals, and weave between obstacles. They use self-supervised learning. The framework finds the 3D apex point for the arm. With a 3D apex point and a task-specific trajectory, they define an arching motion that dynamically manipulates the object. For a fixed apex point they achieved lower performance than a human but with a learned apex point, they were able to achieve much better results.

- 1. Intro:
  - Source Code
  - Tasks: Vaulting, knocking, and weaving
  - They propose to generate trajectories using DJ-GOMP [1], [2].
  - Contributions:
    - 3 novel tasks, vaulting, knocking, and weaving
    - INDy: a self-supervised framework for collecting data and learning a dynamic manipulation policy
    - Related simulation platform
    - Physical experiments with a UR5
  - Model-free, trial and error but they reduce the learning process to just learning the apex point. They use quadratic-programming motion-planning primitive to generate a fast ballistic motion that minimizes the jerk (of the arm) while moving through the apex point.
- 2. Related work: see mind-map
- 3. Problem definition: experiment setup and definitions for spaces and mapping between them. Config space S, for DLO and other objects, obstacles, waypoints, the targets. Observation space O, and mapping of S to O. Tau_a is a complete open-loop trajectory and A is the action space (tau_a in A). SxA->S is a stochastic state transition mapping. Define the goal state. Generates a parametric trajectory.
- 4. Method:
  - A. Hypothesis:
    - Repeatability: reset state by performing a gentle pull on the cable
    - Parameterized trajectory function: They hypothesize that a high-speed min-jerk ballistic manipulator can perform many DLO manipulation tasks by varying the mid-point in location. This way they reduce the trajectory generation to a 2D regression problem, similar to TossingBot [36].
  - B. Deep-Learning Training Algorithm: INDy (Iterative Training for Dynamic Manipulation): Self-supervised method, inputs are a base action for the task a_beta_i, a task-trajectory function f_A_i, and a goal observation O_goal. On failure of the base action, it repeatedly tries random permutations on top of the base action until it completes a task. Self-reset. INDy add success observation-action pairs to D.
  - C. Minimum Jerk Trajectory Generation Function
- 5. Simulator
- 6. Physical Experiments
- 7. Results:
  - A. Comparisons between Simulation and Reality
  - B. Results for Vaulting Task
  - C. Results for Knocking Task
  - D. Results for Weavering Task
  - E. Generalization to Different Cables
  - F. Limitations and Failure Modes
- 8. Conclusion
- Dataset:
- States:
- Sensors: Logitech RGB,
- Hardware: UR5

- Dataset details:
- Additional details:

ICRA 2020

001 - Grasp State Assessment of Deformable Objects Using Visual-Tactile Fusion Perception

- Model: 3D convolution-based visual-tactile fusion deep neural network (implementation)
- Dataset: Grasp State Assessment dataset (GSA dataset)
- Grasping states: sliding, appropriate, and excessive
- The dataset includes grasping and lifting experiments with different widths and forces on 16 deformable objects.
- Sensors: 1080P USB camera as gripper camera, XELA three-axis tactile sensor.
- Hardware: UR3, OnRobot RG2 gripper
- Sensor-Fusion: visual-tactile fusion via fully connected layers
- Dataset details: grasping episodes, each is a grasp and lift trail. Visual frame at 30Hz and tactile samples at 60Hz. 50 to 60 grasp episodes per object (16 deformable objects) where each is a sequence of 30-40 visual frames and 60-80 tactile frames. 13 randomly selected objects were used for training and the remaining three were used for testing.
- Additional grasping experiments with dedicated sections.
- Real-time: yes but not robust.
- Structure
    - Abstract
    - Introduction
    - Related Work
        - Perception
        - XYZ Sense Modality Fusion
        - Grasping with Application-Specific Literature
    - Proposed Method
        - Problem Statement:
        - Model Description:
    - Experiments
        - The Dataset: Based on another dataset or novel. Technical details on robotic and training hardware, sensors and data capture configurations.
        - Performance and Results: Define metrics, and identify factors that directly affect performance for each task (e.g. input resolution, sequence length on detection, tracking and etc).
        - Confusion matrix
        - Other Unique Experiments and Results
    - Conclusion and Future Work

paper title - template

- Brief:
- Model:
- Dataset:
- States:

- Sensors:
- Hardware:
- Sensor-Fusion:
- Dataset details:
- Additional details: