Appendix: A Differentiable Recipe for Learning Visual Non-Prehensile Planar Manipulation

Bernardo Aceituno[1], **Alberto Rodriguez**[1], **Shubham Tulsiani**[2], **Abhinav Gupta**[2], **Mustafa Mukadam**[2]

[1]Massachusetts Institute of Technology, [2]Facebook AI Research

1 Appendix A: Data Generation

In order to train and evaluate this model, we generate a set of random polygonal shapes and random trajectories $\mathbf{r}(t)$. Each object is generated by sampling points along a unit circle, connecting them randomly to form a polygon [1]. This shape is re-scaled to fit within a 20 cm radius. To generate a trajectory we sample a starting pose and a goal pose from an uniform distribution as $\mathbf{r}(0)$, $\mathbf{r}(T) \sim \mathcal{U}(\mathcal{E})$. We linearly interpolate with $T-2$ points between the goal and starting pose, adding a Gaussian noise $d \sim \mathcal{G}(0{,}1\ cm)$ to each position. For each trajectory and shape, we setup $T=5$ and use a video resolution of 50×50.

Once the task is setup, we pose a Mixed-Integer Quadratic Program (MIQP) that find the globally optimal robot finger placements and contact forces $\mathbf{p}_c(t)$,$\lambda_c(t)$ as:

$$\textbf{MIQP:} \qquad \min_{\mathbf{p},\mathbf{\Lambda}} \quad \sum_{t=0}^{T} \lambda_c^T(t) Q_\lambda \lambda_c(t) + \ddot{p}_c^T(t) Q_p \ddot{p}_c(t) \tag{1}$$

$$\text{subject to:} \quad \mathbf{M}\ddot{\mathbf{r}}(t) + \mathbf{G}(\mathbf{r}) = J(\mathbf{r})^T \Lambda(t), \quad \mathbf{p}_c(t) \in \mathbb{F}_c(t), \quad \lambda_c(t) \in \mathcal{FC}_c(t), \quad \lambda_e(t) \in \mathcal{FC}_e(t) \tag{2}$$

This problem is known as Contact-Trajectory Optimization (CTO), and we solve the MIQP using the formulation from [2]. Once the contact-trajectory is generated, we extract the following mechanical parameters:

1. $\mathbf{r}$: object trajectory in $SE(2)$ over T time-steps, represented by one $3 \times T$ matrix.
2. $J(\mathbf{r})(t)$: contact-force Jacobian that maps contact forces into wrenches for object configuration $\mathbf{r}$, represented as a $3 \times 2N$ matrix.
3. $\mathbb{F}_c(t)$: facet where finger $\mathbf{p}_c$ is at contact during time-step t. We represent the facet with two vertices $\phi_1 \in \mathbb{R}^2$,$\phi_2 \in \mathbb{R}^2$.
4. $\mathcal{FC}_c(t)$: friction cone for finger $\mathbf{p}_c$ at time-step t. We represent each friction cone as two rays $\gamma_1 \in \mathbb{R}^2$,$\gamma_2 \in \mathbb{R}^2$.
5. $\mathbf{p}_e(t)$: contact point where external wrenches are aggregated to the object.
6. $\mathcal{FC}_e(t)$: friction cone for the external wrench location at time-step t. We represent this friction cone as two rays $\gamma_1^e \in \mathbb{R}^2$ and $\gamma_2^e \in \mathbb{R}^2$.

All these parameters, along with the solution to CTO are stored and used to train and validate our model. Note that it is difficult to extract these parameters without synthetic data, since real world environments can be very hard to control and measure.

2 Appendix B: Simulated Experiments with OmniPush Shapes

We also study the application of our method towards more realistic data, using shapes and trajectories from the OmniPush dataset [3] adapted to the saggital setting that we consider. We qualitatively demonstrate

5th Conference on Robot Learning (CoRL 2021), London, UK.

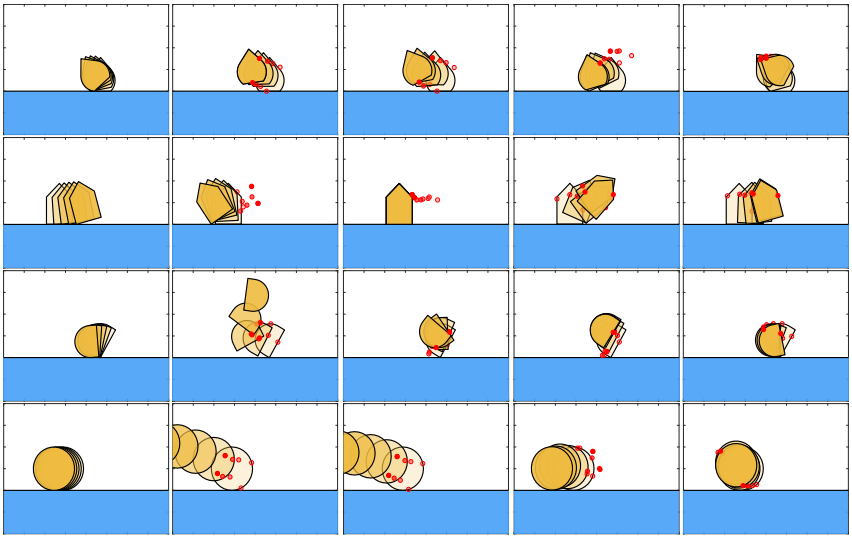

Figure 1: Qualitative examples of each network applied to objects from the OmniPush [3] dataset in a saggital setting. These shapes are significantly smoother than those used for training.

how our model generalizes to this data (consisting of smooth object shapes), without having seen them during training. These results are illustrated in Fig. 1. As before, we observe that our structured architecture performs qualitatively better than less structured approaches.

Real-world videos could directly be utilized as inputs, for example, to a semantic segmentation neural network which then generates the animated style videos as inputs to our proposed architecture and the whole system can also be trained end-to-end. This lies beyond the scope of this paper but would be an interesting future direction.

References

[1] T. Auer. Rpg-heuristics for the generation of random polygons. Citeseer, 1996.

[2] B. Aceituno-Cabezas and A. Rodriguez. A global quasi-dynamic model for contact-trajectory optimization. In *RSS*, 2020.

[3] M. Bauza, F. Alet, Y.-C. Lin, T. Lozano-Pérez, L. P. Kaelbling, P. Isola, and A. Rodriguez. Omnipush: accurate, diverse, real-world dataset of pushing dynamics with rgb-d video. In *IROS*. IEEE, 2019.